ChickenRanging NB4

Network analysis and correlation of network and time warping similarity.

Version: 7.0.

Author: Bernhard Voelkl

Created in: *Mathematica* 10.2

Last modified: 01/03/2018

Data Import

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

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

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

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

In[∘]:=
```
datelist = Union[data[[All, 5]]];
```

In[∘]:=
```
datelist = datelist[[Ordering[Table[AbsoluteTime[{datelist[[i]],
       {"DayName", " ", "Day", "/", "Month", "/", "Year"}}], {i, Length[datelist]}]]]];

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

Functions

In[∘]:=
```
findFriend[i_] := Module[{hen1, pothens, friends},
  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,
   friends = Transpose[{Table[hen1[[1]], {Length[pothens]}], pothens[[All, 1]]}],
   friends = {}
  ];
  friends
 ]
```

In[∘]:=
```
shift = 0;
```

`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 ith 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 henn crosses the antenna within 5 seconds, an empty list is returned, otherwise the function returns a list of edges.

Functions for the standard matrix permutation test

```
In[∘]:= rearrange[mat_] := Module[{p},
   p = RandomPermutation[Length[mat]];
   Permute[Transpose[Apply[Permute, List[Transpose[mat], p]]], p]
  ]
```

rearrange[*mat*] takes as input a square matrix and randomly re-arranges lines and rows (both according to the same random permutation -this ensures that the main diagonal remains empty). The output is the re-arranged matrix.

```
In[∘]:= ppmcc[xmat_, ymat_] := Module[{},
   k = Length[xmat];
   n = (Length[xmat] (Length[xmat] - 1));
   (Total[xmat * ymat, 2] - n * Total[xmat / n, 2] * Total[ymat / n, 2]) /
     ((Total[((xmat - Total[xmat / n, 2])^2) * (1 - IdentityMatrix[k]), 2])
       (Total[((ymat - Total[ymat / n, 2])^2) * (1 - IdentityMatrix[k]), 2]))^0.5 // N]
```

ppmcc[xmat, ymat] is a function that needs two square matrices as input and gives as output the matrix correlation: Pearson's product-moment correlation coefficient, according to Smouse et al 1986 Syst.Zool.

```
In[∘]:= newpermutationR[mat1_, mat2_, permut_] := Module[{},
   statistic = ppmcc[mat1, mat2];
   expectedR = Append[Table[ppmcc[mat1, rearrange[mat2]], {permut - 1}], statistic];
   std = StandardDeviation[expectedR];
   sigma = Abs[Mean[expectedR] - statistic] / std;
   highercounts = Count[expectedR, _? (#1 ≥ statistic &)];
   arr = Graphics[Arrow[{{statistic, -permut/100}, {statistic, 10}}]];
   hist = Histogram[expectedR, PlotLabel -> Grid[{
        {StringJoin["Expected r: ", ToString[Mean[expectedR]], " CI95[", ToString[Mean[
              expectedR] - 1.96 std], ", ", ToString[Mean[expectedR] + 1.96 std], "]"]},
        {StringJoin["Observed r: ", ToString[N[statistic]], ",  N(r+) = ",
          ToString[highercounts], " σ = ", ToString[sigma]]}}]];
   Show[hist, arr, PlotRange → All, ImageSize → 1200, Axes → False,
    Frame → True, FrameStyle → Directive[Thick, Black, 32],
    FrameLabel → {"r²"}, LabelStyle → Directive[24, Black]]]
```

newpermutationR[*mat1,mat2,permut*] is a function that needs three arguments: *mat1* and *mat2*: the matrices to be correlated and *permut*: the number of permutations. It calculates the matrix correlation for *mat1* × *mat2* and for *mat1* with the *permut* times re-arranged matrix *mat2.* It gives as output "r" of *mat1* × *mat2*, *N+*, the number of times that the the permuted matrices had a correlation as large or larger than the observed one and the effect size *sigma* as the difference between observed and expected r in units of standard deviations.

```
In[•]:= newpermutationDataOnly[mat1_, mat2_, permut_] := Module[{},
   statistic = ppmcc[mat1, mat2];
   expectedR = Append[Table[ppmcc[mat1, rearrange[mat2]], {permut - 1}], statistic];
   std = StandardDeviation[expectedR];
   sigma = Abs[Mean[expectedR] - statistic] / std;
   highercounts = Count[expectedR, _? (#1 ≥ statistic &)];
   {N[statistic], expectedR, sigma}
  ]
```

`newpermutationDataOnly[mat1,mat2,permut]` is a function that needs three arguments: *mat1* and *mat2*: the matrices to be correlated and *permut*: the number of permutations. It calculates the matrix correlation for *mat1* × *mat2* and for *mat1* with the *permut* times re-arranged matrix *mat2.* It gives as output "r" of *mat1* × *mat2*, *N+*, the number of times that the the permuted matrices had a correlation as large or larger than the observed one and the effect size *sigma* as the difference between observed and expected r in units of standard deviations.

1. Edge Lists for Overall Networks (summed over 72 days)

For the overall pen network (all days pooled) a weighted edge list is created for each pen based on dyadic associations at pens. A link is added for two hens passing the same antenna (in the same direction) within 5 seconds. (See comment for function `findFriend[i]`).

Edgelist Pen 11

```
edgelistpen11 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 11, datelist[[d]], __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
   AppendTo[edgelistpen11, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]
 ]
]

edgelistpen11 = DeleteCases[Flatten[edgelistpen11, 2], {}];

edgelistpen11 = Sort /@ edgelistpen11;

edgelistpen11 = Tally[edgelistpen11];

edgelistpen11 = DeleteCases[edgelistpen11, {{v1_, v2_}, _} /; v1 == v2];

Save["edgelistpen11", edgelistpen11];
```

Edgelist Pen 12

```
edgelistpen12 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 12, datelist[[d]], __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
   AppendTo[edgelistpen12, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]
 ]
]
```

```
edgelistpen12 = DeleteCases[Flatten[edgelistpen12, 2], {}];
```

```
edgelistpen12 = Sort /@ edgelistpen12;
```

```
edgelistpen12 = Tally[edgelistpen12];
```

```
edgelistpen12 = DeleteCases[edgelistpen12, {{v1_, v2_}, _} /; v1 == v2];
```

```
Save["edgelistpen12", edgelistpen12];
```

Edgelist Pen 13

```
edgelistpen13 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 13, datelist[[d]], __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
   AppendTo[edgelistpen13, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]
 ]
]
```

```
edgelistpen13 = DeleteCases[Flatten[edgelistpen13, 2], {}];
```

```
edgelistpen13 = Sort /@ edgelistpen13;
```

```
edgelistpen13 = Tally[edgelistpen13];
```

```
edgelistpen13 = DeleteCases[edgelistpen13, {{v1_, v2_}, _} /; v1 == v2];
```

```
Save["edgelistpen13", edgelistpen13];
```

Edgelist Pen 14

```
edgelistpen14 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 14, datelist[[d]], __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
   AppendTo[edgelistpen14, ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}]]
 ]
]
```

```
edgelistpen14 = DeleteCases[Flatten[edgelistpen14, 2], {}];
```

```
edgelistpen14 = Sort /@ edgelistpen14;
```

```
edgelistpen14 = Tally[edgelistpen14];
```

```
edgelistpen14 = DeleteCases[edgelistpen14, {{v1_, v2_}, _} /; v1 == v2];

Save["edgelistpen14", edgelistpen14];
```

2. Association Indices for Overall Networks (summed over 72 days)

Pen 11

```
allhens11 = Cases[allhens, {_, "11", __}];

edgelistpen11 = Get["edgelistpen11"];

data11 = Cases[data, {_, _, _, 11, __}];

indivobservations =
  Table[Length[Cases[data11, {allhens11[[i, 3]], __}]], {i, Length[allhens11]}];

n = Length[allhens11];

amat = Table[0, {n}, {n}];

For[i = 1, i ≤ n, i++,
 For[j = 1, j ≤ n, j++,
  pair = Cases[edgelistpen11, {{allhens11[[i, 3]], allhens11[[j, 3]]}, _}];
  If[Length[pair] > 0,
   amat[[i, j]] = pair[[1, 2]]
  ]
 ]
]
```

The association index (AI) is calculated as: X(AB)/(X(A) + X(B) - X(AB))

```
amat = amat + Transpose[amat];

aimat = N[Table[If[indivobservations[[i]] + indivobservations[[j]] > 0,
     amat[[i, j]] / (indivobservations[[i]] + indivobservations[[j]] - amat[[i, j]]), "NA"],
    {i, Length[allhens11]}, {j, Length[allhens11]}]];

Save["aimat11", aimat];

Export["aimat11.csv", aimat];
```

Pen 12

```
distmat = Get["distmat12"];

dmhead = distmat[[1]];

allhens12 = Cases[allhens, {_, "12", __}];

edgelistpen12 = Get["edgelistpen12"];

data12 = Cases[data, {_, _, _, 12, __}];
```

```
indivobservations =
  Table[Length[Cases[data12, {allhens12[[i, 3]], __}]], {i, Length[allhens12]}];

n = Length[allhens12];

amat = Table[0, {n}, {n}];

For[i = 1, i ≤ n, i++,
 For[j = 1, j ≤ n, j++,
  pair = Cases[edgelistpen12, {{allhens12[[i, 3]], allhens12[[j, 3]]}, _}];
  If[Length[pair] > 0,
   amat[[i, j]] = pair[[1, 2]]
  ]
 ]
]

(*Association index: X(AB)/( X(A) + X(B) - X(AB))*)

amat = amat + Transpose[amat];

aimat = N[Table[If[indivobservations[[i]] + indivobservations[[j]] > 0,
     amat[[i, j]] / (indivobservations[[i]] + indivobservations[[j]] - amat[[i, j]]), "NA"],
    {i, Length[allhens12]}, {j, Length[allhens12]}]];

Save["aimat12", aimat];

Export["aimat12.csv", aimat];
```

Pen 13

```
distmat = Get["distmat13"];

dmhead = distmat[[1]];

allhens13 = Cases[allhens, {_, "13", __}];

edgelistpen13 = Get["edgelistpen13"];

data13 = Cases[data, {_, _, _, 13, __}];

indivobservations =
  Table[Length[Cases[data13, {allhens13[[i, 3]], __}]], {i, Length[allhens13]}];

n = Length[allhens13];

amat = Table[0, {n}, {n}];

For[i = 1, i ≤ n, i++,
 For[j = 1, j ≤ n, j++,
  pair = Cases[edgelistpen13, {{allhens13[[i, 3]], allhens13[[j, 3]]}, _}];
  If[Length[pair] > 0,
   amat[[i, j]] = pair[[1, 2]]
  ]
 ]
]

(*Association index: X(AB)/( X(A) + X(B) - X(AB))*)

amat = amat + Transpose[amat];
```

```
aimat = N[Table[If[indivobservations⟦i⟧ + indivobservations⟦j⟧ > 0,
      amat⟦i, j⟧ / (indivobservations⟦i⟧ + indivobservations⟦j⟧ - amat⟦i, j⟧), "NA"],
     {i, Length[allhens13]}, {j, Length[allhens13]}]];

Save["aimat13", aimat];

Export["aimat13.csv", aimat];
```

Pen 14

```
distmat = Get["distmat14"];

dmhead = distmat⟦1⟧;

allhens14 = Cases[allhens, {_, "14", __}];

edgelistpen14 = Get["edgelistpen14"];

data14 = Cases[data, {_, _, _, 14, __}];

indivobservations =
  Table[Length[Cases[data14, {allhens14⟦i, 3⟧, __}]], {i, Length[allhens14]}];

n = Length[allhens14];

amat = Table[0, {n}, {n}];

For[i = 1, i ≤ n, i++,
 For[j = 1, j ≤ n, j++,
  pair = Cases[edgelistpen14, {{allhens14⟦i, 3⟧, allhens14⟦j, 3⟧}, _}];
  If[Length[pair] > 0,
   amat⟦i, j⟧ = pair⟦1, 2⟧
  ]
 ]
]

(*Association index: X(AB)/( X(A) + X(B) - X(AB))*)

amat = amat + Transpose[amat];

aimat = N[Table[If[indivobservations⟦i⟧ + indivobservations⟦j⟧ > 0,
      amat⟦i, j⟧ / (indivobservations⟦i⟧ + indivobservations⟦j⟧ - amat⟦i, j⟧), "NA"],
     {i, Length[allhens14]}, {j, Length[allhens14]}]];

Save["aimat14", aimat];

Export["aimat14.csv", aimat];
```

3. Network Descriptors for Overall Networks (summed over 72 days)

Pen 11

```
edgelistpen11 = Get["edgelistpen11"];

aimat = Get["aimat11"];
```

```
pen11aigraph = WeightedAdjacencyGraph[aimat //. {"NA" → 0, 0. → 0},
   EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];
```

```
pen11aimultigraph = AdjacencyGraph[Round[(aimat //. {"NA" → 0, 0. → 0}) 10 000],
   DirectedEdges → True, EdgeStyle → Arrowheads[0]];
```

```
N[GraphDensity[pen11aimultigraph]] (*Graphdensity*)
```

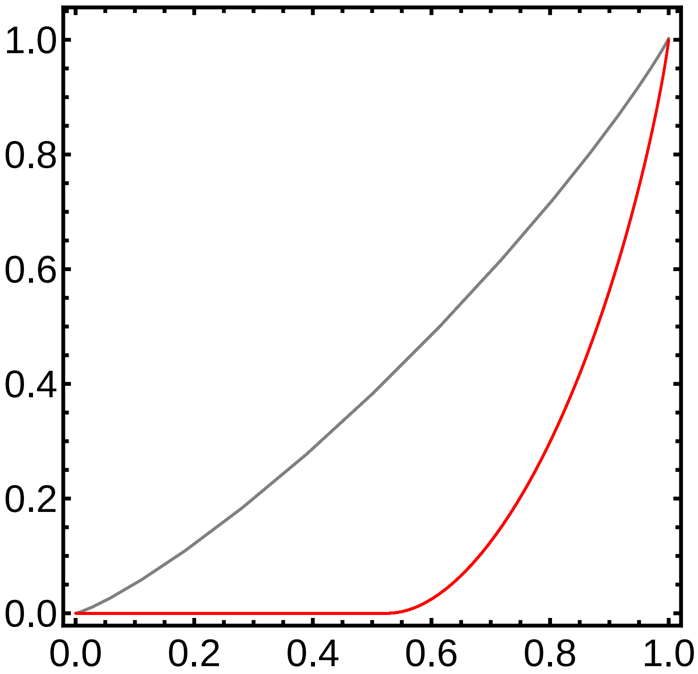
```
0.942714
```

```
max = Total[edgelistpen11[[All, 2]]] (*total number of links*)
```

```
134 060
```

```
n = Length[aimat];
k = n (n - 1);
```

```
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]],
  ListPlot[Transpose[{Range[k] / k,
     PadLeft[Accumulate[Sort[edgelistpen11[[All, 2]]]], k, 0] / max}], PlotStyle → Red]]
```



Expected (grey) and observed (red) CDF for edge weights

```
GraphAssortativity[pen11aigraph, FindGraphCommunities[pen11aigraph]]
```

```
0.0931133
```

```
btw11 = BetweennessCentrality[pen11aimultigraph];
```

```
prc11 = PageRankCentrality[pen11aimultigraph, 0.1];
```

```
lsc11 = StatusCentrality[pen11aimultigraph];
```

```
gc11 = FindGraphCommunities[pen11aigraph];
```

```
vertexlist11 = VertexList[pen11aigraph];

cms11 = Table[Position[gc11, vertexlist11[[i]]][[1, 1]], {i, Length[vertexlist11]}];

gc11 // Length (*Number of graph communities*)
```

5

Pen 12

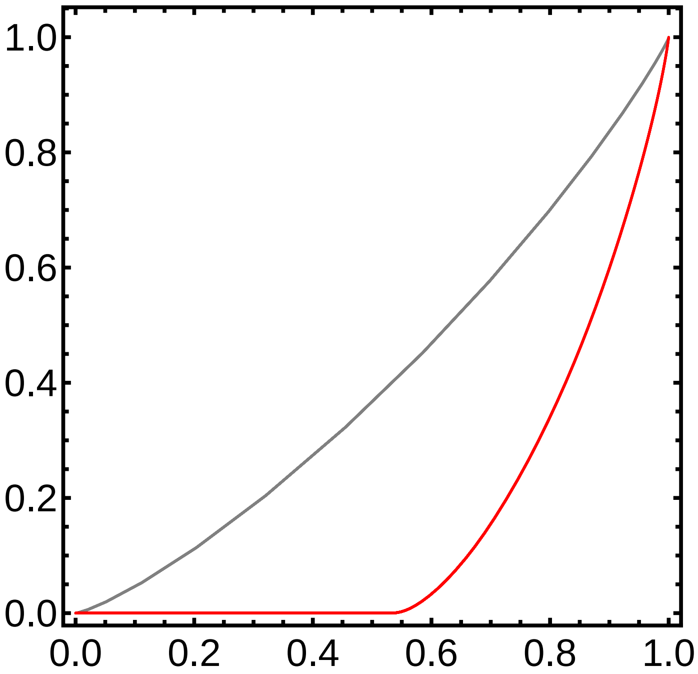
```
edgelistpen12 = Get["edgelistpen12"];

aimat = Get["aimat12"];

pen12aigraph = WeightedAdjacencyGraph[aimat //. {"NA" → 0, 0. → 0},
   EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];

pen12aimultigraph = AdjacencyGraph[Round[(aimat //. {"NA" → 0, 0. → 0}) 10000],
   DirectedEdges → True, EdgeStyle → Arrowheads[0]];

N[GraphDensity[pen12aimultigraph]] (*Graphdensity*)
```

0.931277

```
max = Total[edgelistpen12[[All, 2]]] (*total number of links*)
```

95 219

```
n = Length[aimat];
k = n (n - 1);

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]],
  ListPlot[Transpose[{Range[k] / k,
     PadLeft[Accumulate[Sort[edgelistpen12[[All, 2]]]], k, 0] / max}], PlotStyle → Red]]
```



```
(*Expected (grey) and oberved (red) CDF for edeweights *)
```

```
GraphAssortativity[pen12aigraph, FindGraphCommunities[pen12aigraph]]
```

```
0.0844989
```

```
btw12 = BetweennessCentrality[pen12aimultigraph];
```

```
prc12 = PageRankCentrality[pen12aimultigraph, 0.1];
```

```
lsc12 = StatusCentrality[pen12aimultigraph];
```

```
gc12 = FindGraphCommunities[pen12aigraph];
vertexlist12 = VertexList[pen12aigraph];
cms12 = Table[Position[gc12, vertexlist12[[i]]][[1, 1]], {i, Length[vertexlist12]}];
```

```
gc12 // Length
```

```
4
```

Pen 13

```
edgelistpen13 = Get["edgelistpen13"];
```

```
aimat = Get["aimat13"];
```

```
pen13aigraph = WeightedAdjacencyGraph[aimat //. {"NA" → 0, 0. → 0},
   EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];
```

```
pen13aimultigraph = AdjacencyGraph[Round[(aimat //. {"NA" → 0, 0. → 0}) 10000],
   DirectedEdges → True, EdgeStyle → Arrowheads[0]];
```

```
N[GraphDensity[pen13aimultigraph]] (*Graphdensity*)
```

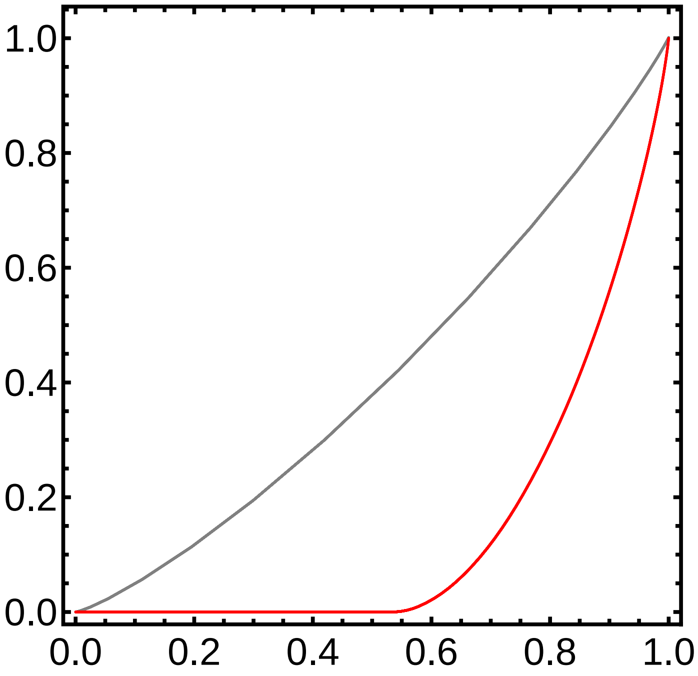
```
0.924708
```

```
max = Total[edgelistpen13[[All, 2]]] (*total number of links*)
```

```
114310
```

```
n = Length[aimat];
k = n (n - 1);
```

```
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]],
 ListPlot[Transpose[{Range[k] / k,
    PadLeft[Accumulate[Sort[edgelistpen13[[All, 2]]]], k, 0] / max}], PlotStyle → Red]]
```




```
(*Expected (grey) and oberved (red) CDF for edeweights *)

GraphAssortativity[pen13aigraph, FindGraphCommunities[pen13aigraph]]

0.109329

btw13 = BetweennessCentrality[pen13aimultigraph];

prc13 = PageRankCentrality[pen13aimultigraph, 0.1];

lsc13 = StatusCentrality[pen13aimultigraph];

gc13 = FindGraphCommunities[pen13aigraph];
vertexlist13 = VertexList[pen13aigraph];
cms13 = Table[Position[gc13, vertexlist13[[i]]][[1, 1]], {i, Length[vertexlist13]}];

gc13 // Length

5
```

Pen 14

```
edgelistpen14 = Get["edgelistpen14"];

aimat = Get["aimat14"];

pen14aigraph = WeightedAdjacencyGraph[aimat //. {"NA" → 0, 0. → 0},
   EdgeWeight, DirectedEdges → True, EdgeStyle → Arrowheads[0]];
```

```
pen14aimultigraph = AdjacencyGraph[Round[(aimat //. {"NA" → 0, 0. → 0}) 10000],
    DirectedEdges → True, EdgeStyle → Arrowheads[0]];
```

```
N[GraphDensity[pen14aimultigraph]] (*Graphdensity*)
```

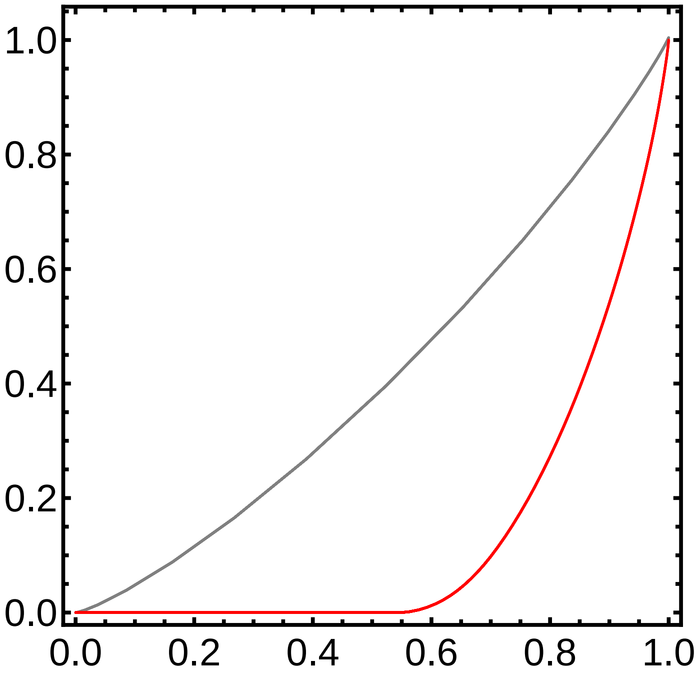
```
0.901792
```

```
max = Total[edgelistpen14[[All, 2]]] (*total number of links*)
```

```
101334
```

```
n = Length[aimat];
k = n (n - 1);
```

```
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]],
  ListPlot[Transpose[{Range[k] / k,
     PadLeft[Accumulate[Sort[edgelistpen14[[All, 2]]]], k, 0] / max}], PlotStyle → Red]]
```




```
(*Expected (grey) and oberved (red) CDF for edeweights *)
```

```
GraphAssortativity[pen14aigraph, FindGraphCommunities[pen14aigraph]]
```

```
0.120954
```

```
btw14 = BetweennessCentrality[pen14aimultigraph];
```

```
prc14 = PageRankCentrality[pen14aimultigraph, 0.1];
```

```
lsc14 = StatusCentrality[pen14aimultigraph];
```

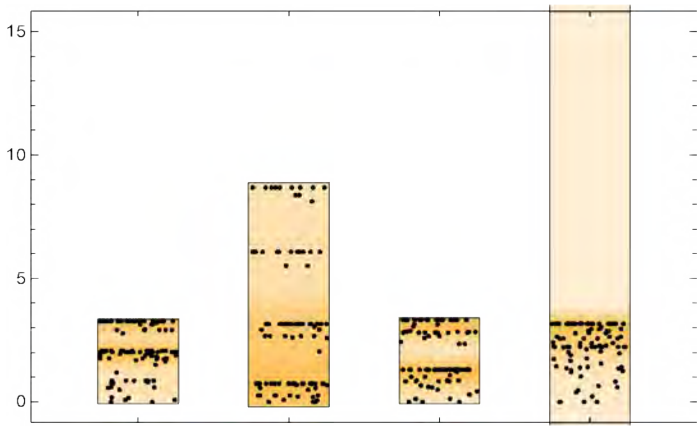
```
gc14 = FindGraphCommunities[pen14aigraph];
vertexlist14 = VertexList[pen14aigraph];
cms14 = Table[Position[gc14, vertexlist14[[i]]][[1, 1]], {i, Length[vertexlist14]}];
```

```
gc14 // Length
```

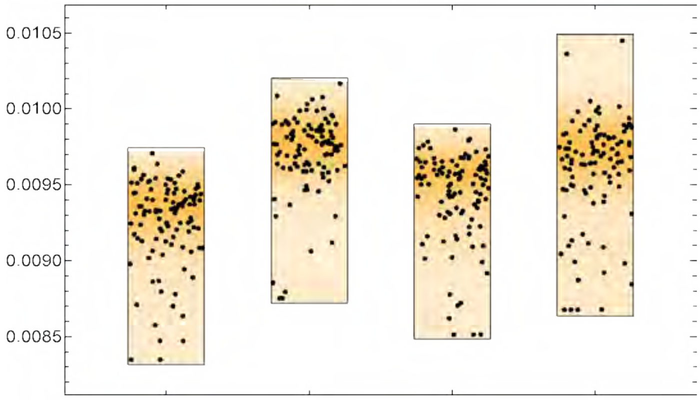
6

Individual Measures

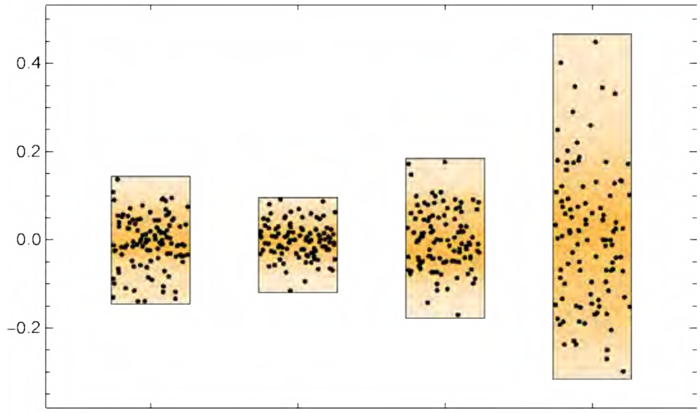
```
DistributionChart[{btw11, btw12, btw13, btw14},
 PlotRange → {All, {0, 15}}, ChartElementFunction → "PointDensity"]
```




```
DistributionChart[{prc11, prc12, prc13, prc14}, ChartElementFunction → "PointDensity"]
```




```
DistributionChart[{lsc11, lsc12, lsc13, lsc14}, ChartElementFunction → "PointDensity"]
```



Network measure Matrix

```
networkmeasures =
  {Join[allhens11[[All, 1]], allhens12[[All, 1]], allhens13[[All, 1]], allhens14[[All, 1]]],
   Join[Table[11, {Length[vertexlist11]}], Table[12, {Length[vertexlist12]}],
    Table[13, {Length[vertexlist13]}], Table[14, {Length[vertexlist14]}]],
   Join [btw11, btw12, btw13, btw14], Join [prc11, prc12, prc13, prc14],
   Join [lsc11, lsc12, lsc13, lsc14] //. Indeterminate → 0,
   Join[0.1 cms11 + 11, 0.1 cms12 + 12 , 0.1 cms13 + 13, 0.1 cms14 + 14]};

networkmeasures = Prepend[Transpose[networkmeasures],
   {"Bird", "Pen", "Betweenness", "Centrality", "Status", "Community"}];

Export["networkmeasuresnew.csv", networkmeasures]

networkmeasuresnew.csv
```

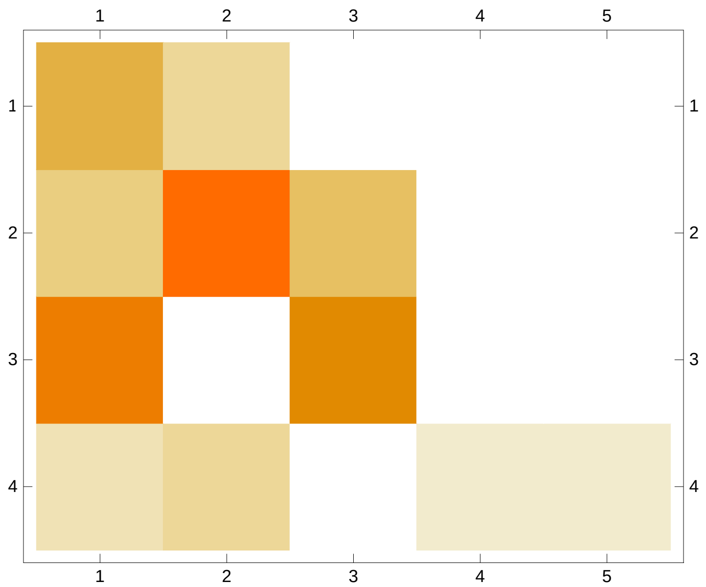
Correlating NW measures with DTW Clusters

Pen 11

```
cl11 = Import["cl11.csv", "table", FieldSeparators → ";"];
```

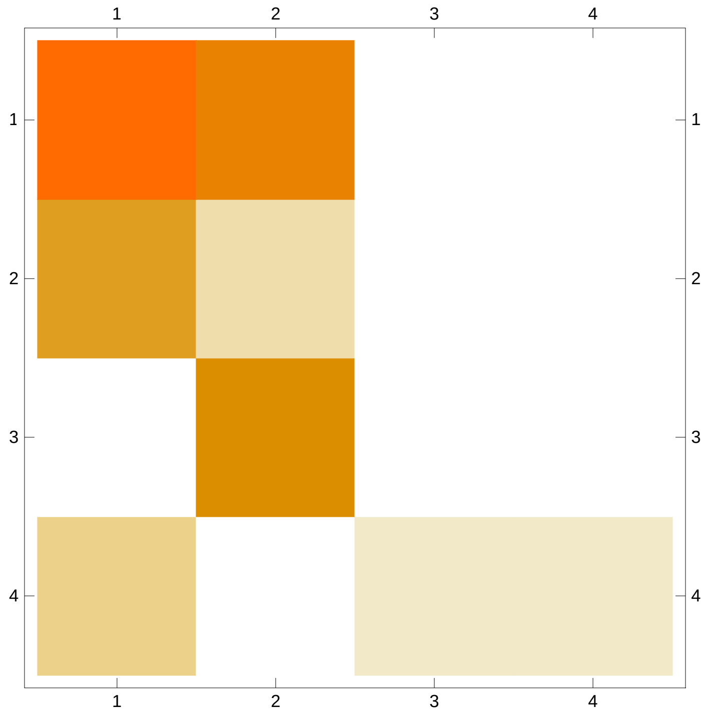
Classifications based on dynamic time warping (computed in R) are imported as .csv files.

```
dtwclusters11 = Table[Cases[cl11, {_, ci}][[All, 1]], {ci, 1, 4}]

ngmodules11 = Table[Cases[networkmeasures, {__, 11 + 0.1 ci}][[All, 1]], {ci, 1, 5}];

MatrixPlot[Table[
  Table[Length[Intersection[dtwclusters11[[t]], ngmodules11[[u]]]], {u, 1, 5}], {t, 1, 4}]]
```



Pen 12

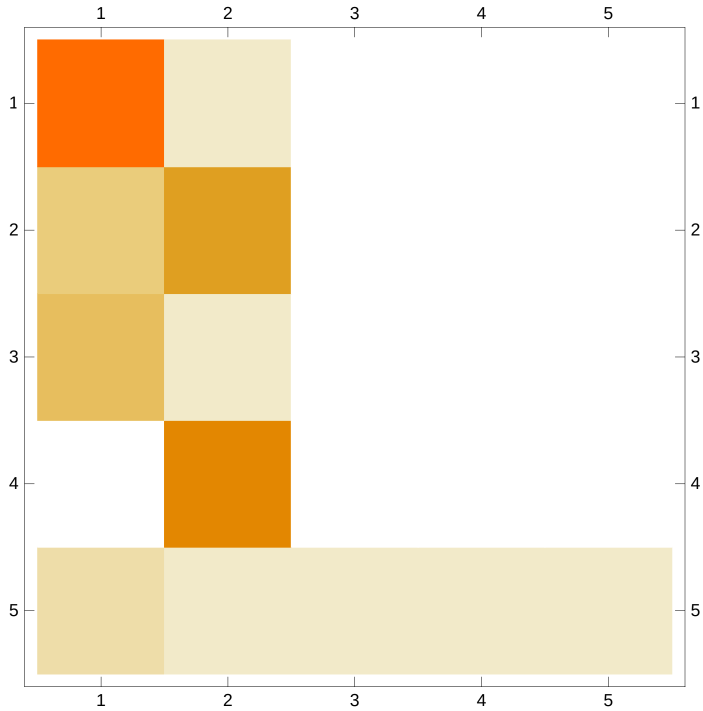
```
cl12 = Import["cl12.csv", "table", FieldSeparators → ";"];

dtwclusters12 = Table[Cases[cl12, {_, ci}][[All, 1]], {ci, 1, 4}]

ngmodules12 = Table[Cases[networkmeasures, {__, 12 + 0.1 ci}][[All, 1]], {ci, 1, 4}];

MatrixPlot[Table[
  Table[Length[Intersection[dtwclusters12[[t]], ngmodules12[[u]]]], {u, 1, 4}], {t, 1, 4}]]
```



Pen 13

```
cl13 = Import["cl13.csv", "table", FieldSeparators → ";"];

dtwclusters13 = Table[Cases[cl13, {_, ci}][[All, 1]], {ci, 1, 5}]

ngmodules13 = Table[Cases[networkmeasures, {__, 13 + 0.1 ci}][[All, 1]], {ci, 1, 5}]
```

```
MatrixPlot[Table[
  Table[Length[Intersection[dtwclusters13[[t]], ngmodules13[[u]]]], {u, 1, 5}], {t, 1, 5}]]
```



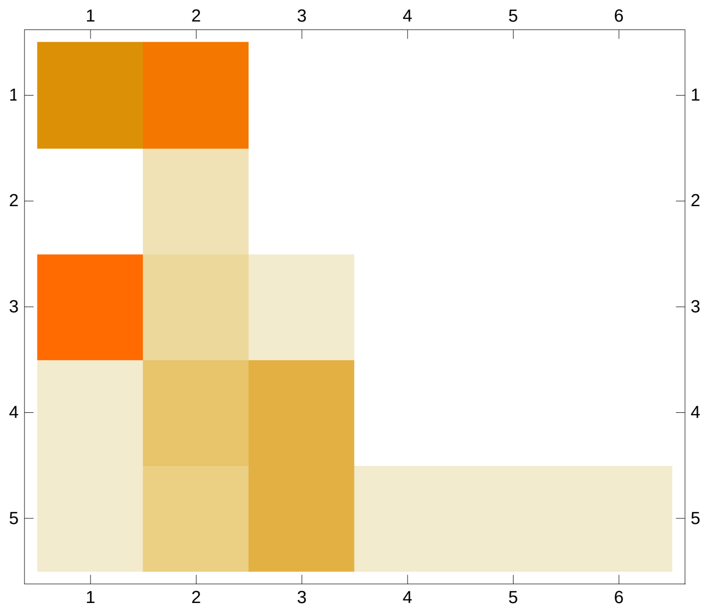
Pen 14

```
cl14 = Import["cl14.csv", "table", FieldSeparators → ";"];
```

```
dtwclusters14 = Table[Cases[cl14, {_, ci}][[All, 1]], {ci, 1, 5}]
```

```
ngmodules14 = Table[Cases[networkmeasures, {__, 14 + 0.1 ci}][[All, 1]], {ci, 1, 6}];
```

```
MatrixPlot[Table[
  Table[Length[Intersection[dtwclusters14[[t]], ngmodules14[[u]]]], {u, 1, 6}], {t, 1, 5}]]
```



Combined probability

G statistics for the contingency tables were calculated in the R script “GTestClustersModules”. The combined probability for all four pens is summarized below

```
chisquared =
 -2 (Log[7.1 × 10^-14] + Log[6.3 × 10^-10] + Log[1 × 10^-13] + Log[2.1 × 10^-7])
```

193.542

```
N[Probability[x >= chisquared, x ≈ ChiSquareDistribution[8]]]
```

1.46365×10^{-37}

Daily Networks

```
dayofyear = {158, 192, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206,
   207, 208, 209, 210, 211, 213, 220, 222, 223, 224, 225, 226, 227, 228, 229,
   230, 234, 235, 236, 237, 238, 239, 240, 242, 243, 244, 245, 246, 247, 249,
   250, 251, 252, 253, 254, 255, 256, 257, 264, 265, 266, 267, 268, 269, 270,
   271, 272, 273, 274, 276, 277, 278, 279, 280, 281, 283, 284, 285, 286, 289};
```

```
shift = 0;
```

Pen 11

```
edgelistpen11 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 11, datelist⟦d⟧, __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
     dailyedgelist =
    DeleteCases[Tally[Sort /@ DeleteCases[Flatten[ParallelTable[findFriend[i],
          {i, 1, Length[pen] - 1}], 1], {}]], {{v1_, v2_}, _} /; v1 == v2];
    AppendTo[edgelistpen11, dailyedgelist],
   Print[StringJoin["No data for this day: ", ToString[d]]]
  ]
]
```

No data for this day: 68

```
Save["dailyedgelistpen11", edgelistpen11];
```

```
Length[edgelistpen11]
```

71

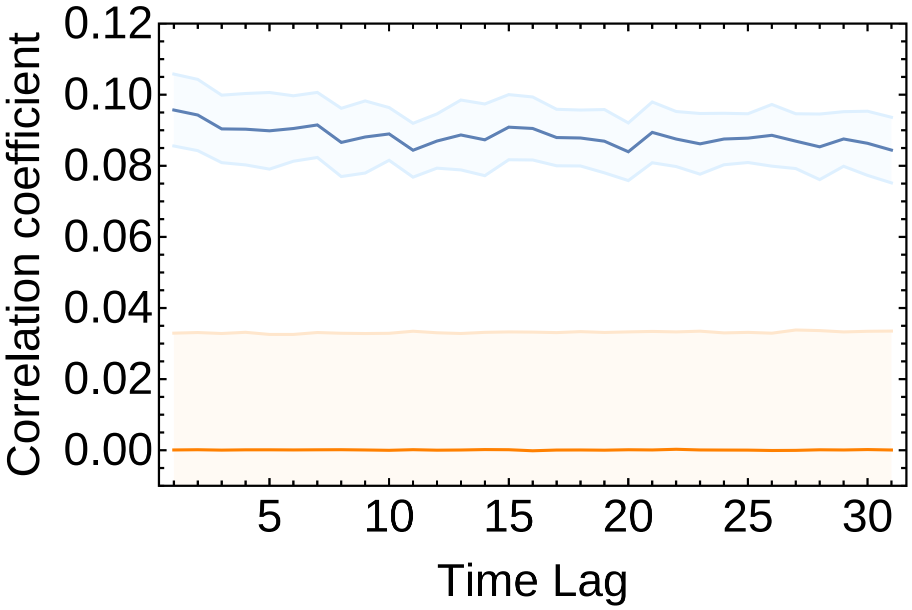
edgelistpen11 is a nested list with each of the 71 sub-lists being the edgelist for a single day.

```
edgelistpen11 = Get["dailyedgelistpen11"];
```

```
wms = Table[WeightedAdjacencyMatrix[Graph[Cases[allhens, {_, "11", __}]⟦All, 3⟧,
      Apply[UndirectedEdge, edgelistpen11⟦i, All, 1⟧, 1],
      EdgeWeight → edgelistpen11⟦i, All, 2⟧]], {i, Length[edgelistpen11]}];
```

This creates for pen 11 for each day a weighted adjacency matrix

```
days = Length[wms];

newacorrSN11 = Table[
   Table[newpermutationDataOnly[wms⟦q⟧, wms⟦i⟧, 1000], {i, q + 1, days}], {q, 1, 40}];

Save["newacorrSN11", newacorrSN11]

obsl = Table[Mean[Table[newacorrSN11⟦i, lag, 1⟧, {i, 40}]], {lag, 31}];

ciobs =
  1.96 * (Table[StandardDeviation[Table[newacorrSN11⟦i, lag, 1⟧, {i, 40}]], {lag, 31}] /
     Sqrt[40]);

exm = Table[Mean[Table[Mean[newacorrSN11⟦i, lag, 2⟧], {i, 40}]], {lag, 31}];

exci = Table[Mean[Table[Sort[newacorrSN11⟦i, lag, 2⟧]⟦975⟧, {i, 40}]], {lag, 31}];

SNconsistency11 = Show[ListLinePlot[exci, PlotRange → {All, {-0.01, 0.12}},
   PlotStyle → LightOrange, Filling → -0.01, Frame → True,
   FrameStyle → Thickness[0.003], FrameTicksStyle → Directive[Black, 24],
   FrameLabel → {{"Correlation coefficient", ""}, {"Time Lag", ""}},
   LabelStyle -> Directive[Black, 24]], ListLinePlot[exm, PlotStyle → Orange],
  ListLinePlot[{obsl + ciobs, obsl - ciobs}, PlotStyle → LightBlue, Filling → {1 → {2}}],
  ListLinePlot[obsl], ImageSize → 800]
```



This plot shows the observed (blue) and expected (orange) correlation coefficients of days t = 1 to 40 with days with timelags t+1 to t+31.

```
Export["SNconsistency11.tif", SNconsistency11]
```

SNconsistency11.tif

Pen 12

```
edgelistpen12 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 12, datelist[[d]], __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
     dailyedgelist =
   DeleteCases[Tally[Sort /@ DeleteCases[Flatten[ParallelTable[findFriend[i],
          {i, 1, Length[pen] - 1}], 1], {}]], {{v1_, v2_}, _} /; v1 == v2];
    AppendTo[edgelistpen12, dailyedgelist],
  Print[StringJoin["No data for this day: ", ToString[d]]]
 ]
]
```

```
Save["dailyedgelistpen12", edgelistpen12];
```

```
edgelistpen12 = Get["dailyedgelistpen12"];
```

```
Length[edgelistpen12]
```

72

```
wms = Table[WeightedAdjacencyMatrix[Graph[Cases[allhens, {_, "12", __}][[All, 3]],
      Apply[UndirectedEdge, edgelistpen12[[i, All, 1]], 1],
      EdgeWeight → edgelistpen12[[i, All, 2]]]], {i, Length[edgelistpen12]}];
```

```
days = Length[wms];
```

```
newacorrSN12 = Table[ParallelTable[
    newpermutationDataOnly[wms[[q]], wms[[i]], 1000], {i, q + 1, days}], {q, 1, 40}];
```

```
Save["newacorrSN12", newacorrSN12]
```

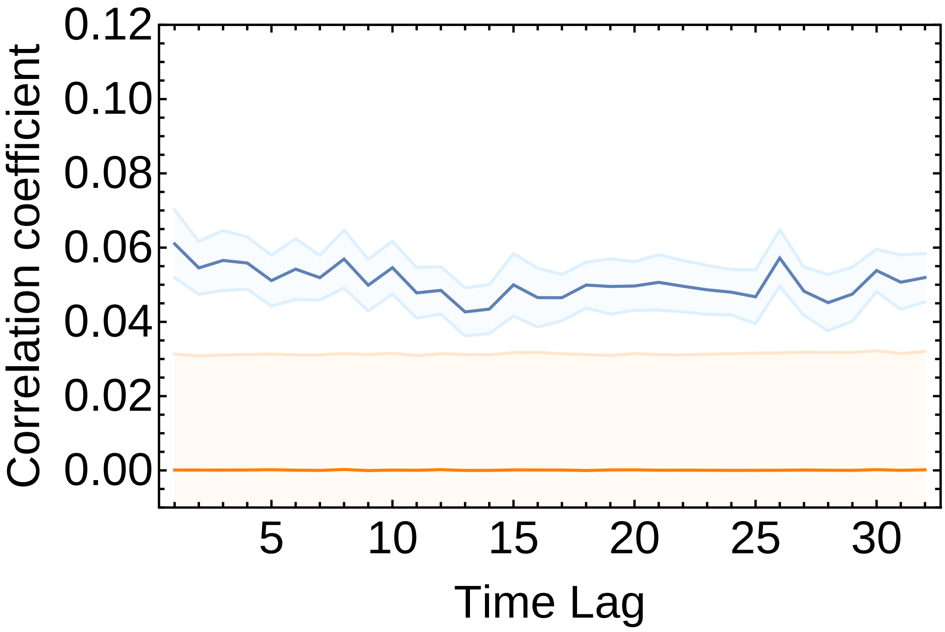
```
obsl = Table[Mean[Table[newacorrSN12[[i, lag, 1]], {i, 40}]], {lag, 32}]
```

```
ciobs =
 1.96 * (Table[StandardDeviation[Table[newacorrSN12[[i, lag, 1]], {i, 40}]], {lag, 32}] /
    Sqrt[40])
```

```
exm = Table[Mean[Table[Mean[newacorrSN12[[i, lag, 2]]], {i, 40}]], {lag, 32}]
```

```
exci = Table[Mean[Table[Sort[newacorrSN12[[i, lag, 2]]][[975]], {i, 40}]], {lag, 32}]
```

```
SNconsistency12 = Show[ListLinePlot[exci, PlotRange → {All, {-0.01, 0.12}},
    PlotStyle → LightOrange, Filling → -0.01, Frame → True,
    FrameStyle → Thickness[0.003], FrameTicksStyle → Directive[Black, 24],
    FrameLabel → {{"Correlation coefficient", ""}, {"Time Lag", ""}},
    LabelStyle -> Directive[Black, 24]], ListLinePlot[exm, PlotStyle → Orange],
  ListLinePlot[{obsl + ciobs, obsl - ciobs}, PlotStyle → LightBlue, Filling → {1 → {2}}],
  ListLinePlot[obsl], ImageSize → 800]
```




```
Export["SNconsistency12.tif", SNconsistency12]
```

SNconsistency12.tif

Pen 13

```
edgelistpen13 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 13, datelist[[d]], __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
     dailyedgelist =
    DeleteCases[Tally[Sort /@ DeleteCases[Flatten[ParallelTable[findFriend[i],
          {i, 1, Length[pen] - 1}], 1], {}]], {{v1_, v2_}, _} /; v1 == v2];
     AppendTo[edgelistpen13, dailyedgelist],
   Print[StringJoin["No data for this day: ", ToString[d]]]
  ]
]
```

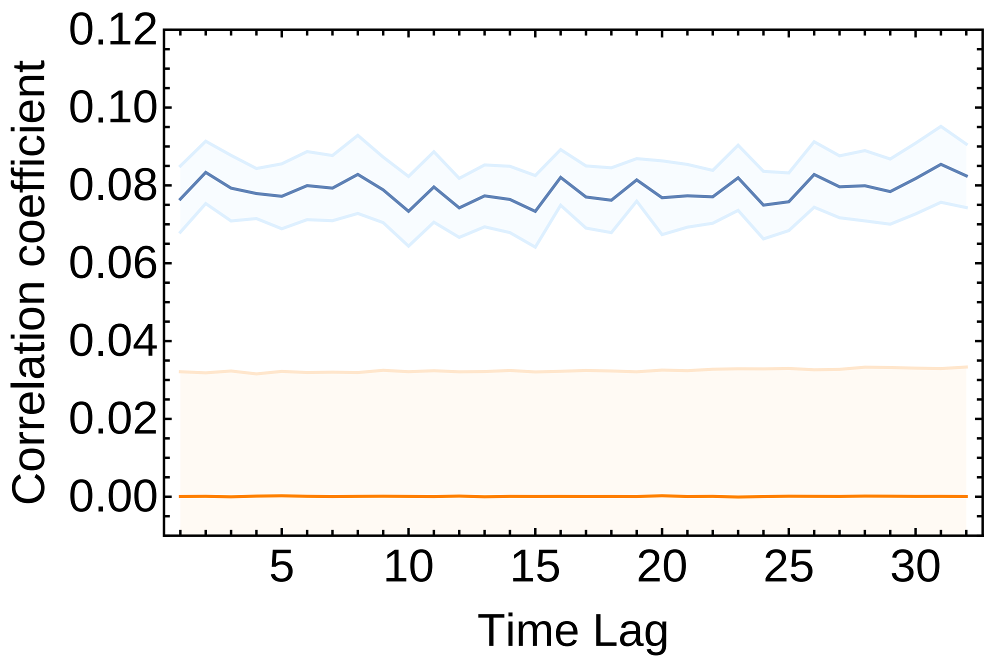
```
Save["dailyedgelistpen13", edgelistpen13];
```

```
edgelistpen13 = Get["dailyedgelistpen13"];
```

```
Length[edgelistpen13]
```

72

```
wms = Table[WeightedAdjacencyMatrix[Graph[Cases[allhens, {_, "13", __}][[All, 3]],
      Apply[UndirectedEdge, edgelistpen13[[i, All, 1]], 1],
      EdgeWeight → edgelistpen13[[i, All, 2]]]], {i, Length[edgelistpen13]}];

days = Length[wms];

newacorrSN13 = Table[ParallelTable[
    newpermutationDataOnly[wms[[q]], wms[[i]], 1000], {i, q + 1, days}], {q, 1, 40}];

Save["newacorrSN13", newacorrSN13]

obsl = Table[Mean[Table[newacorrSN13[[i, lag, 1]], {i, 40}]], {lag, 32}]

ciobs =
 1.96 * (Table[StandardDeviation[Table[newacorrSN13[[i, lag, 1]], {i, 40}]], {lag, 32}] /
    Sqrt[40])

exm = Table[Mean[Table[Mean[newacorrSN13[[i, lag, 2]]], {i, 40}]], {lag, 32}]

exci = Table[Mean[Table[Sort[newacorrSN13[[i, lag, 2]]][[975]], {i, 40}]], {lag, 32}]

SNconsistency13 = Show[ListLinePlot[exci, PlotRange → {All, {-0.01, 0.12}},
   PlotStyle → LightOrange, Filling → -0.01, Frame → True,
   FrameStyle → Thickness[0.003], FrameTicksStyle → Directive[Black, 24],
   FrameLabel → {{"Correlation coefficient", ""}, {"Time Lag", ""}},
   LabelStyle -> Directive[Black, 24]], ListLinePlot[exm, PlotStyle → Orange],
  ListLinePlot[{obsl + ciobs, obsl - ciobs}, PlotStyle → LightBlue, Filling → {1 → {2}}],
  ListLinePlot[obsl], ImageSize → 800]
```




```
Export["SNconsistency13.tif", SNconsistency13]
```

SNconsistency13.tif

Pen 14

```
edgelistpen14 = {};
For[d = 1, d ≤ Length[datelist], d++,
 pen = Cases[data, {_, _, _, 14, datelist⟦d⟧, __}];
 pen = Sort[pen, #1[[6]] < #2[[6]] &];
 If[Length[pen] > 1,
     dailyedgelist =
   DeleteCases[Tally[Sort /@ DeleteCases[Flatten[ParallelTable[findFriend[i],
          {i, 1, Length[pen] - 1}], 1], {}]], {{v1_, v2_}, _} /; v1 == v2];
    AppendTo[edgelistpen14, dailyedgelist],
   Print[StringJoin["No data for this day: ", ToString[d]]]
 ]
]
```

```
Save["dailyedgelistpen14", edgelistpen14];
```

```
edgelistpen14 = Get["dailyedgelistpen14"];
```

```
Length[edgelistpen14]
```

72

```
wms = Table[WeightedAdjacencyMatrix[Graph[Cases[allhens, {_, "14", __}]⟦All, 3⟧,
      Apply[UndirectedEdge, edgelistpen14⟦i, All, 1⟧, 1],
      EdgeWeight → edgelistpen14⟦i, All, 2⟧]], {i, Length[edgelistpen14]}];
```

```
days = Length[wms];
```

```
newacorrSN14 = Table[ParallelTable[
     newpermutationDataOnly[wms⟦q⟧, wms⟦i⟧, 1000], {i, q + 1, days}], {q, 1, 40}];
```

```
Save["newacorrSN14", newacorrSN14]
```

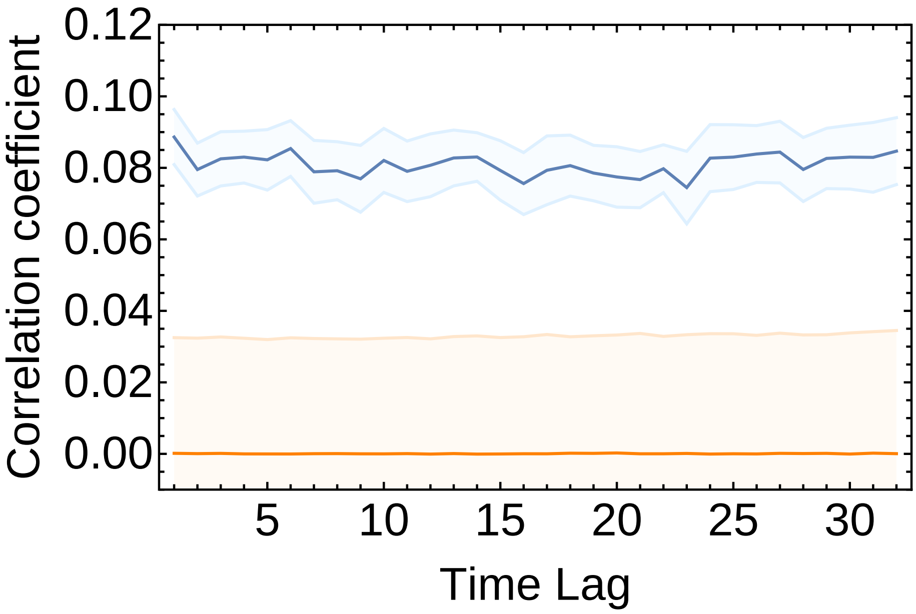
```
obsl = Table[Mean[Table[newacorrSN14⟦i, lag, 1⟧, {i, 40}]], {lag, 32}]
```

```
ciobs =
 1.96 * (Table[StandardDeviation[Table[newacorrSN14⟦i, lag, 1⟧, {i, 40}]], {lag, 32}] /
     Sqrt[40])
```

```
exm = Table[Mean[Table[Mean[newacorrSN14⟦i, lag, 2⟧], {i, 40}]], {lag, 32}]
```

```
exci = Table[Mean[Table[Sort[newacorrSN14⟦i, lag, 2⟧]⟦975⟧, {i, 40}]], {lag, 32}]
```

```
SNconsistency14 = Show[ListLinePlot[exci, PlotRange → {All, {-0.01, 0.12}},
   PlotStyle → LightOrange, Filling → -0.01, Frame → True,
   FrameStyle → Thickness[0.003], FrameTicksStyle → Directive[Black, 24],
   FrameLabel → {{"Correlation coefficient", ""}, {"Time Lag", ""}},
   LabelStyle -> Directive[Black, 24]], ListLinePlot[exm, PlotStyle → Orange],
  ListLinePlot[{obsl + ciobs, obsl - ciobs}, PlotStyle → LightBlue, Filling → {1 → {2}}],
  ListLinePlot[obsl], ImageSize → 800]
```




```
Export["SNconsistency14.tif", SNconsistency14]
SNconsistency14.tif
```

Correlation AI and DTW

Pen 11

```
dm11 = Get["distmat11"];
```

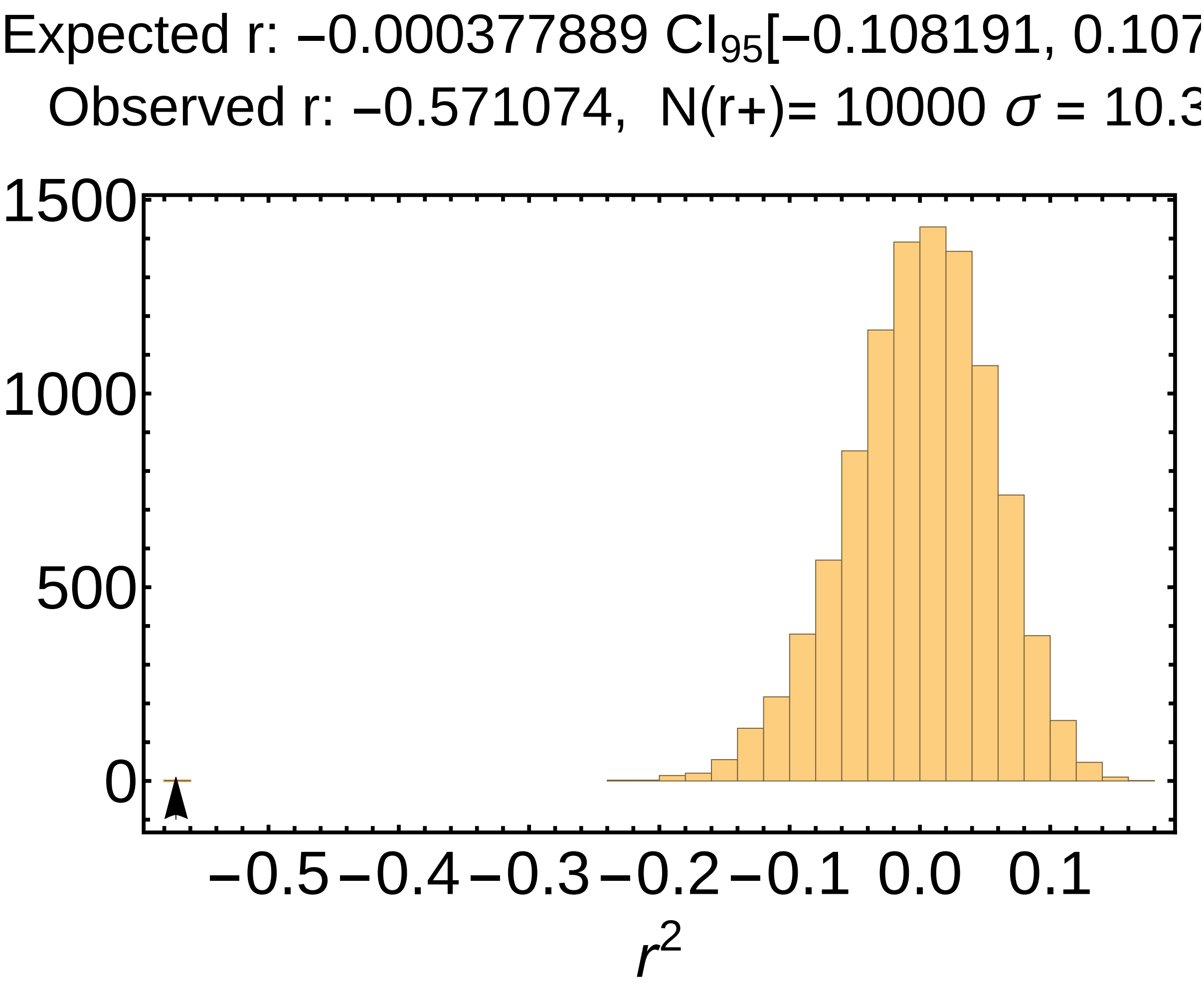
Importing dynamic time warping distance matrices.

```
dmhead = dm11[[1]];

chicks11 = Table[StringDrop[dmhead[[i]], -3], {i, 2, Length[dmhead], 72}];

allhens11 = Cases[allhens, {_, "11", __}];

n = Length[chicks11];

dmat = Table[0, {n}, {n}];
```

```
For[i = 1, i ≤ Length[allhens11], i++,
 For[j = 1, j ≤ Length[allhens11], j++,
  chicki = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks11⟦i⟧ ~~ ___]]]];
  chickj = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks11⟦j⟧ ~~ ___]]]];
  dmat⟦i, j⟧ = N[Mean[Table[dm11⟦chicki⟦u⟧, chickj⟦u⟧⟧, {u, Length[chicki]}]]]
 ]
]

Save["dmat11", dmat];
```

This is re-formatting the dtw distance matrix (same format as adjacency matrix from social networks).

```
aimat11 = Get["aimat11"];

correlpen11 = newpermutationR[aimat11 //. "NA" → 0, dmat, 10000]
```



Correlation of overall (over all days for a pen) adjacency matric with dtw distance matrix using Mantel matrix permutation.

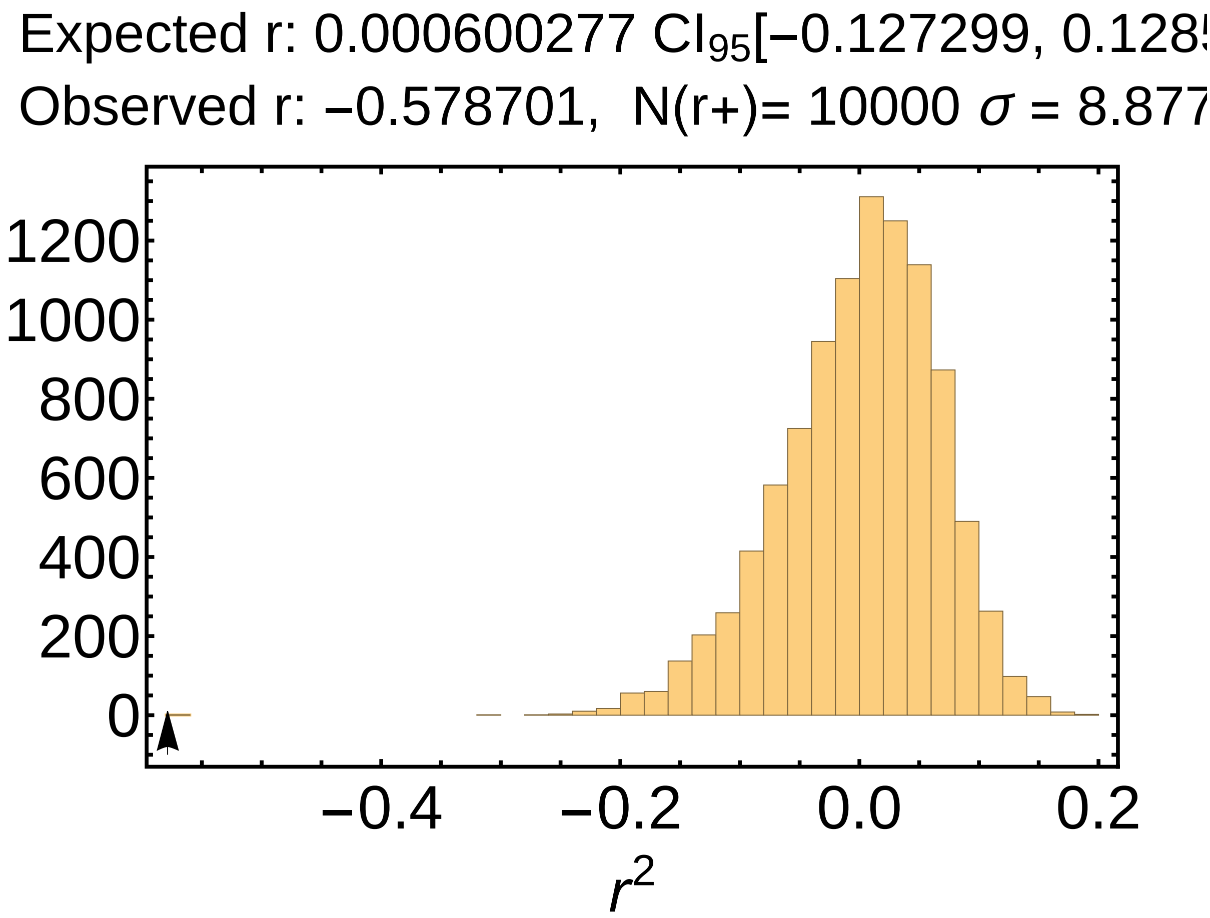
```
Export["DTWxAIcorPen11.tiff", correlpen11]
```

```
DTWxAIcorPen11.tiff
```

Pen 12

```
dm12 = Get["distmat12"];

dmhead = dm12[[1]];

chicks12 = Table[StringDrop[dmhead[[i]], -3], {i, 2, Length[dmhead], 72}];

allhens12 = Cases[allhens, {_, "12", __}];

n = Length[chicks12];

dmat = Table[0, {n}, {n}];

For[i = 1, i ≤ Length[allhens12], i++,
 For[j = 1, j ≤ Length[allhens12], j++,
  chicki = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks12[[i]] ~~ ___]]]];
  chickj = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks12[[j]] ~~ ___]]]];
  dmat[[i, j]] = N[Mean[Table[dm12[[chicki[[u]], chickj[[u]]]], {u, Length[chicki]}]]]
 ]
]

Save["dmat12", dmat];

dmat = Get["dmat12"];

aimat12 = Get["aimat12"];
```

```
correlpen12 = newpermutationR[aimat12 //. "NA" → 0, dmat, 10000]
```




```
Export["DTWxAIcorPen12.tiff", correlpen12]
```

```
DTWxAIcorPen12.tiff
```

Pen 13

```
dm13 = Get["distmat13"];
```

```
dmhead = dm13[[1]];
```

```
chicks13 = Table[StringDrop[dmhead[[i]], -3], {i, 2, Length[dmhead], 72}];
```

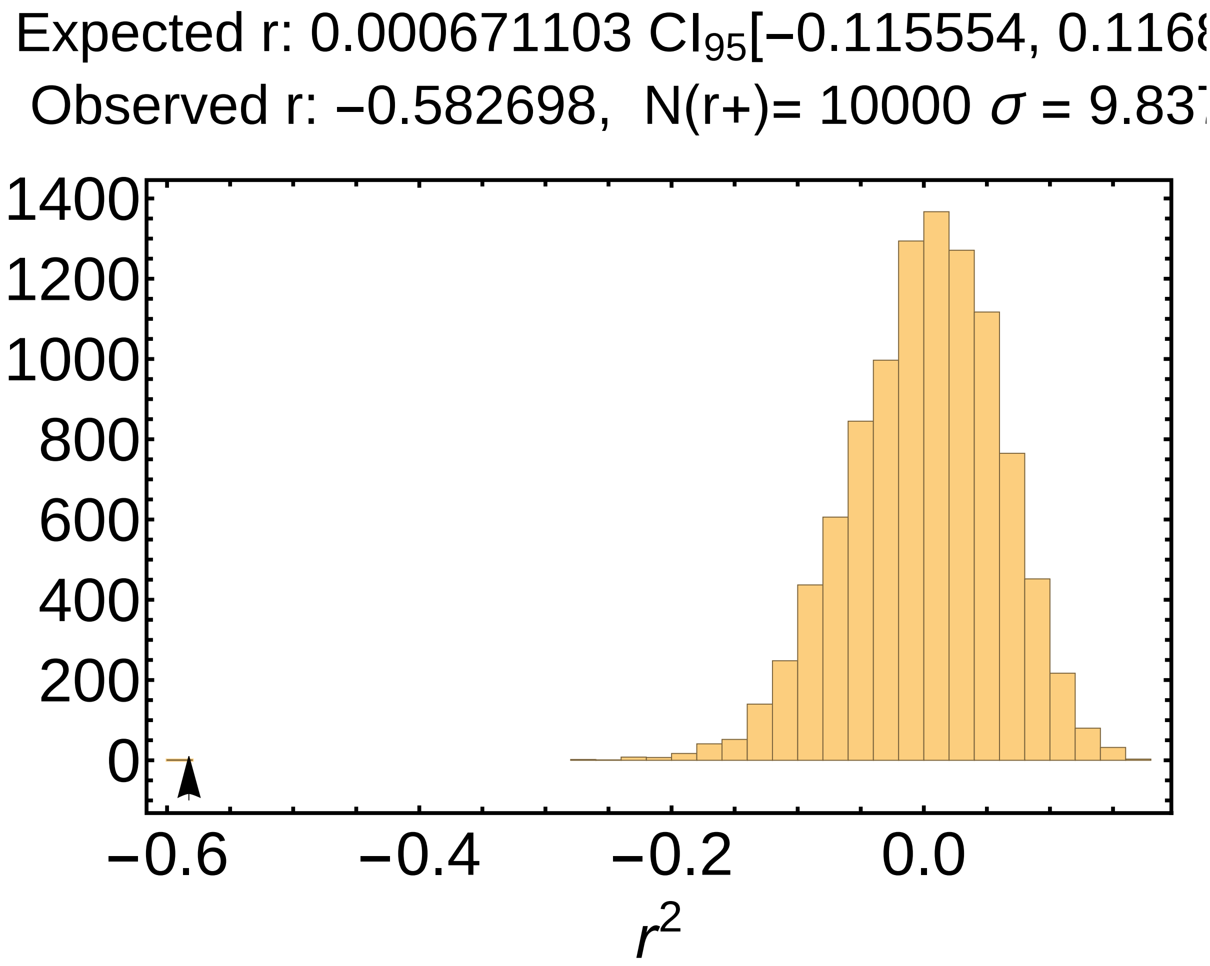
```
allhens13 = Cases[allhens, {_, "13", __}];
```

```
n = Length[chicks13];
```

```
dmat = Table[0, {n}, {n}];
```

```
For[i = 1, i ≤ Length[allhens13], i++,
 For[j = 1, j ≤ Length[allhens13], j++,
  chicki = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks13[[i]] ~~ ___]]]];
  chickj = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks13[[j]] ~~ ___]]]];
  dmat[[i, j]] = N[Mean[Table[dm13[[chicki[[u]], chickj[[u]]]], {u, Length[chicki]}]]]
 ]
]
```

```
Save["dmat13", dmat];

dmat = Get["dmat13"];

aimat13 = Get["aimat13"];

correlpen13 = newpermutationR[aimat13 //. "NA" → 0, dmat, 10000]
```




```
Export["DTWxAIcorPen13.tiff", correlpen13]
DTWxAIcorPen13.tiff
```

Pen 14

```
dm14 = Get["distmat14"];

dmhead = dm14[[1]];

chicks14 = Table[StringDrop[dmhead[[i]], -3], {i, 2, Length[dmhead], 72}];

allhens14 = Cases[allhens, {_, "14", __}];

n = Length[chicks14];

dmat = Table[0, {n}, {n}];
```

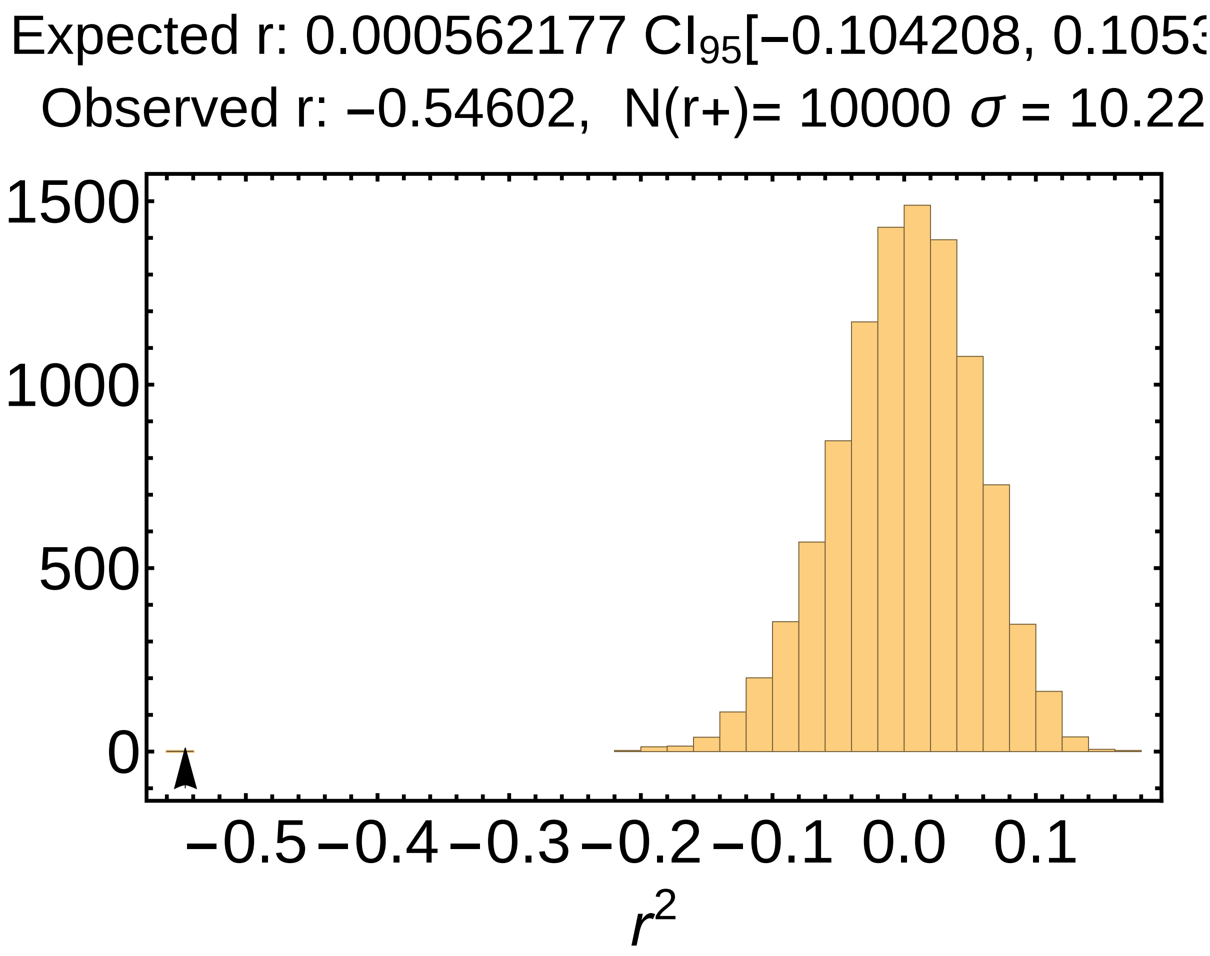
```
For[i = 1, i ≤ Length[allhens14], i++,
 For[j = 1, j ≤ Length[allhens14], j++,
  chicki = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks14[[i]] ~~ ___]]]];
  chickj = Quiet[Flatten[Position[dmhead, x_ /; StringMatchQ[x, chicks14[[j]] ~~ ___]]]];
  dmat[[i, j]] = N[Mean[Table[dm14[[chicki[[u]], chickj[[u]]]], {u, Length[chicki]}]]]
 ]
]
```

```
Save["dmat14", dmat];
```

```
dmat = Get["dmat14"];
```

```
aimat14 = Get["aimat14"];
```

```
correlpen14 = newpermutationR[aimat14 //. "NA" → 0, dmat, 10000]
```




```
Export["DTWxAIcorPen14.tiff", correlpen14]
```

```
DTWxAIcorPen14.tiff
```

DTW Distane Matrix

Here matrix correlations (Pearson product moment correlations) between AI and DTW distance are calculated for each day.

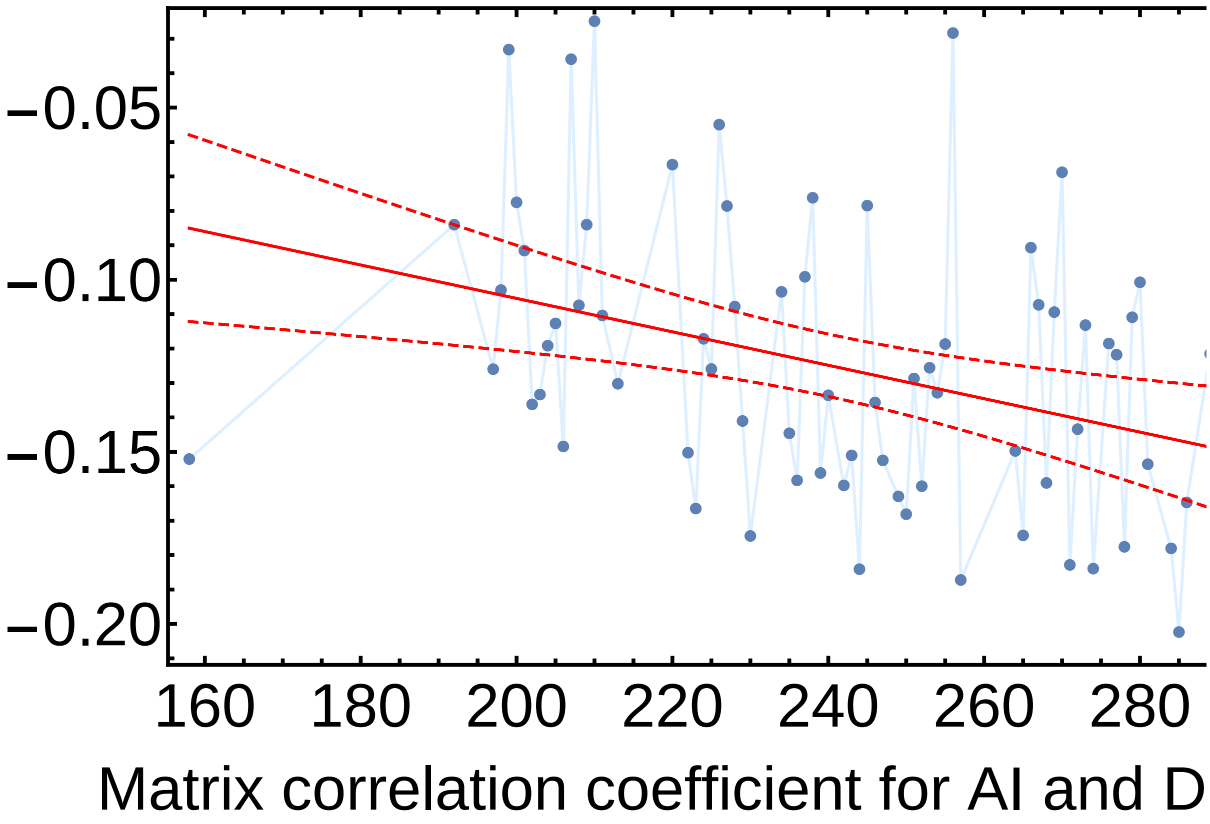
Pen 11

```
distmat = Get["distmat11"];

hens = Cases[allhens, {_, "11", __}];

dailyedgelistpen = Get["dailyedgelistpen11"];

order = {51, 13, 25, 34, 16, 55, 66, 44, 5, 26, 35, 18, 57, 69,
   47, 8, 29, 10, 12, 62, 41, 4, 24, 32, 15, 54, 65, 43, 17, 56, 68, 46, 7,
   28, 37, 59, 71, 38, 1, 21, 30, 50, 61, 40, 3, 23, 31, 14, 53, 64, 67, 45,
   6, 27, 36, 19, 58, 70, 48, 9, 20, 11, 49, 60, 39, 2, 22, 52, 63, 42, 33} ;
 (*dailyedgelistpen and dtw table is not in chronological order*)

wms = Table[WeightedAdjacencyMatrix[
    Graph[hens[[All, 3]], Apply[UndirectedEdge, dailyedgelistpen[[i, All, 1]], 1],
     EdgeWeight → dailyedgelistpen[[i, All, 2]]]], {i, Length[dailyedgelistpen]}];

dm = Delete[
   Table[distmat[[Table[i, {i, 2 + u, 7777 + u, 72}], Table[i, {i, 2 + u, 7777 + u, 72}]]],
    {u, 0, 71}], 13]; (*Oct 10 is deleted because only zeros*)

mcs = Table[ppmcc[dm[[order[[i]]]], wms[[order[[i]]]]], {i, 1, 71}];

(*for pen 11 no data for day 68, so this one was dropped*)

mcs = Transpose[{Drop[dayofyear, {68}], mcs}];

reg = LinearModelFit[mcs, {1, x}, x]

FittedModel[ -0.00832358 - 0.000485527 x ]
```

A linear regression model with day of the year as predictor for the Pearson product moment correlation between DTW and AI.

```
line = reg["BestFit"]

-0.00832358 - 0.000485527 x

cis = reg["MeanPredictionBands"];
```

```
similSNTW = Show[ListLinePlot[mcs, PlotStyle → LightBlue],
  ListPlot[mcs], Plot[line, {x, 158, 289}, PlotStyle → Red],
  Plot[cis[[1]], {x, 158, 289}, PlotStyle → {Red, Dashed}],
  Plot[cis[[2]], {x, 158, 289}, PlotStyle → {Red, Dashed}], PlotRange → All,
  ImageSize → 800, Axes → False, Frame → True, FrameStyle → Directive[Thick, Black, 32],
  FrameLabel → {"Matrix correlation coefficient for AI and DTW"},
  LabelStyle → Directive[24, Black]]
```




```
reg["ANOVATable"]
```

	DF	SS	MS	F-Statistic	P-Value
x	1	0.0142053	0.0142053	9.70249	0.00268009
Error	69	0.101022	0.00146409		
Total	70	0.115227			

```
reg["AdjustedRSquared"]
```

```
0.110575
```

Time (as linear covariate) explains 11% of the overall variation in the correlation coefficients.

```
Export["AIxDTWoverTimePen11.tif", similSNTW]
```

```
AIxDTWoverTimePen11.tif
```

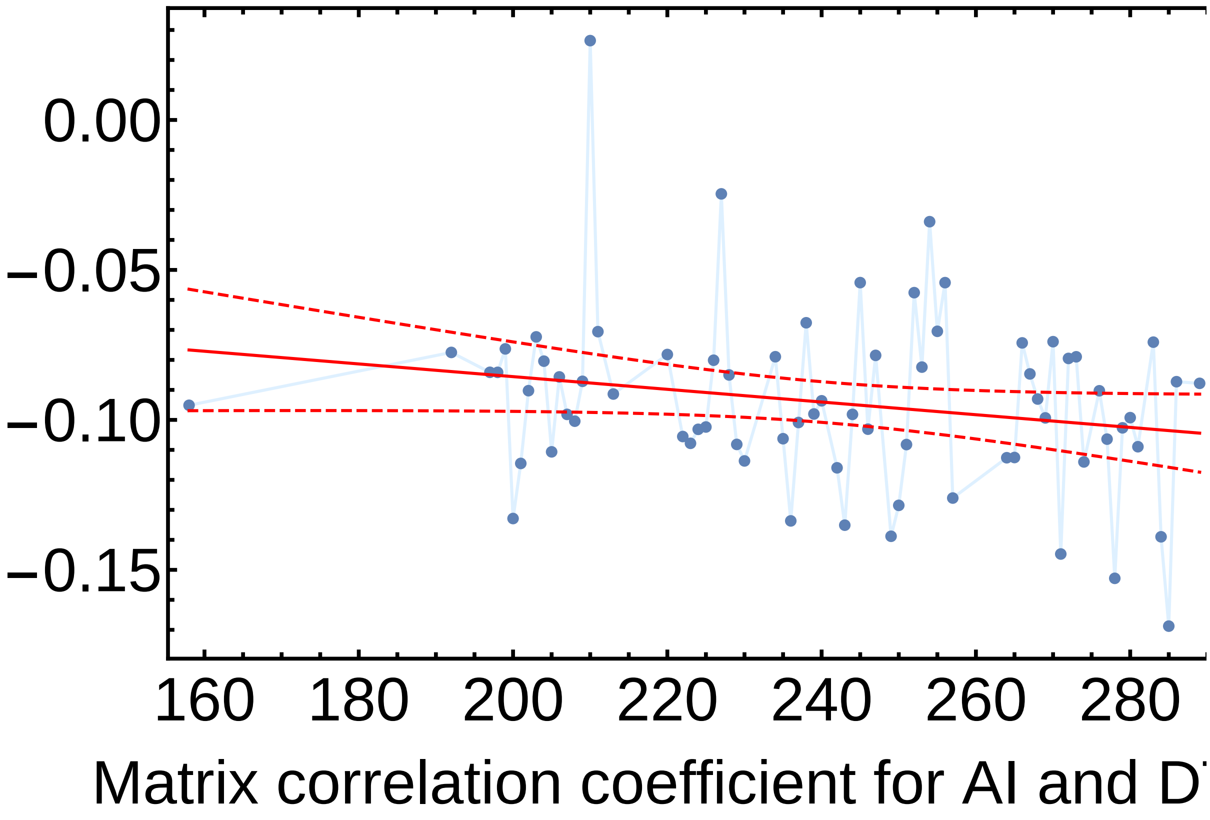
Pen 12

```
distmat = Get["distmat12"];
```

```
hens = Cases[allhens, {_, "12", __}];
```

```
dailyedgelistpen = Get["dailyedgelistpen12"];

datelist = Get["datelist"]; (*sorted*)

order = {52, 14, 26, 35, 17, 56, 67, 45, 5, 27, 36, 19, 58, 70, 48,
   8, 30, 10, 12, 63, 42, 4, 25, 33, 16, 55, 66, 44, 18, 57, 69, 47, 7, 29,
   38, 60, 72, 39, 1, 22, 31, 51, 62, 41, 3, 24, 32, 15, 54, 65, 68, 46, 6,
   28, 37, 20, 59, 71, 49, 9, 21, 11, 50, 61, 40, 2, 23, 13, 53, 64, 43, 34};
 (*dailyedgelistpen and dtw table is not in chronological order*)

wms = Table[WeightedAdjacencyMatrix[
     Graph[hens[[All, 3]], Apply[UndirectedEdge, dailyedgelistpen[[i, All, 1]], 1],
      EdgeWeight → dailyedgelistpen[[i, All, 2]]]], {i, Length[dailyedgelistpen]}];

dm = Table[distmat[[Table[i, {i, 2 + u, 7417 + u, 72}],
     Table[i, {i, 2 + u, 7417 + u, 72}]]], {u, 0, 71}];

mcs = Table[ppmcc[dm[[order[[i]]]], wms[[order[[i]]]]], {i, 1, 72}];

mcs = Transpose[{dayofyear, mcs}];

reg = LinearModelFit[mcs, {1, x}, x]

FittedModel[ -0.0432518 - 0.000211642 x ]

line = reg["BestFit"]

-0.0432518 - 0.000211642 x

cis = reg["MeanPredictionBands"];
```

```
similSNTW = Show[ListLinePlot[mcs, PlotStyle → LightBlue, PlotRange → All],
  ListPlot[mcs], Plot[line, {x, 158, 289}, PlotStyle → Red],
  Plot[cis[[1]], {x, 158, 289}, PlotStyle → {Red, Dashed}],
  Plot[cis[[2]], {x, 158, 289}, PlotStyle → {Red, Dashed}], PlotRange → All,
  ImageSize → 800, Axes → False, Frame → True, FrameStyle → Directive[Thick, Black, 32],
  FrameLabel → {"Matrix correlation coefficient for AI and DTW"},
  LabelStyle → Directive[24, Black]]
```




```
reg["ANOVATable"]
```

	DF	SS	MS	F-Statistic	P-Value
x	1	0.00278007	0.00278007	3.3546	0.0712725
Error	70	0.0580113	0.000828733		
Total	71	0.0607914			

```
reg["AdjustedRSquared"]
```

```
0.0320989
```

```
Export["AIxDTWoverTimePen12.tif", similSNTW]
```

```
AIxDTWoverTimePen12.tif
```

Pen 13

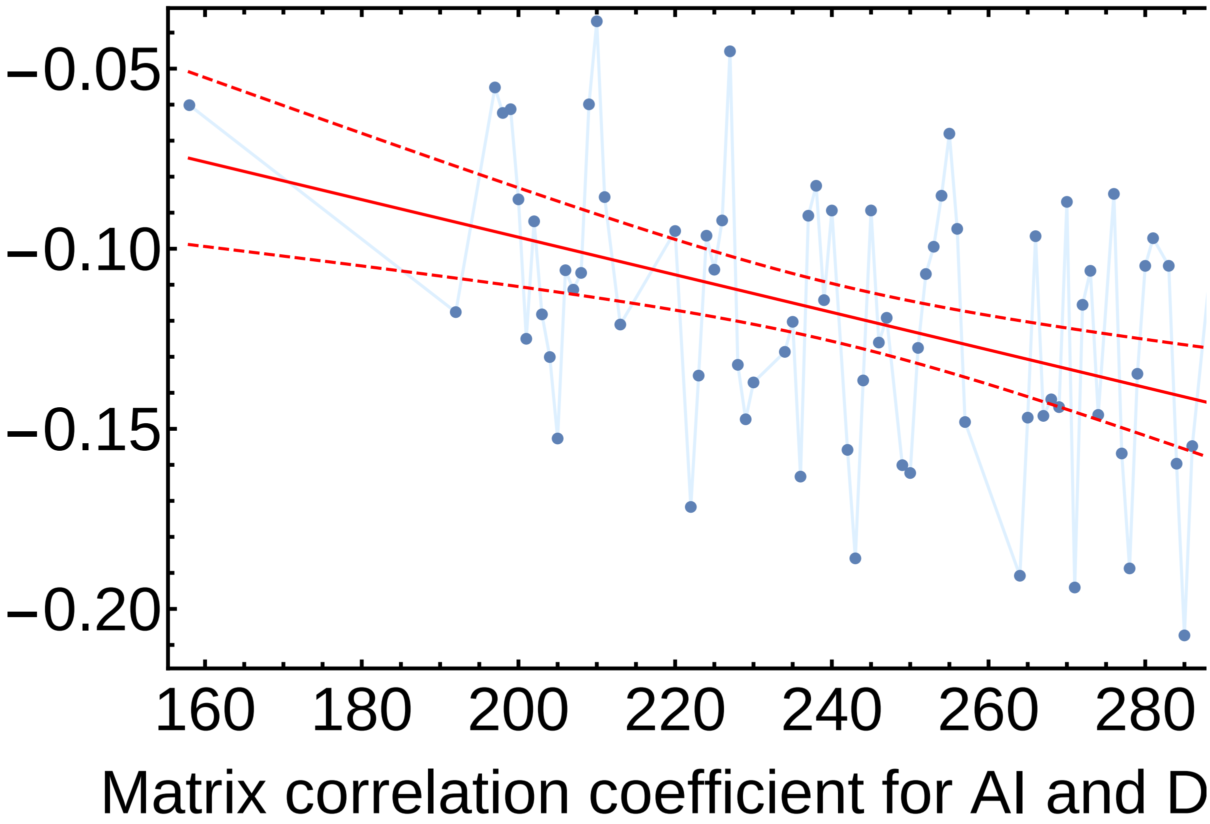
```
distmat = Get["distmat13"];
```

```
hens = Cases[allhens, {_, "13", __}];
```

```
dailyedgelistpen = Get["dailyedgelistpen13"];
```

```
datelist = Get["datelist"]; (*sorted*)

order = {52, 14, 26, 35, 17, 56, 67, 45, 5, 27, 36, 19, 58, 70, 48,
   8, 30, 10, 12, 63, 42, 4, 25, 33, 16, 55, 66, 44, 18, 57, 69, 47, 7, 29,
   38, 60, 72, 39, 1, 22, 31, 51, 62, 41, 3, 24, 32, 15, 54, 65, 68, 46, 6,
   28, 37, 20, 59, 71, 49, 9, 21, 11, 50, 61, 40, 2, 23, 13, 53, 64, 43, 34};
 (*dailyedgelistpen and dtw table is not in chronological order*)

wms = Table[WeightedAdjacencyMatrix[
    Graph[hens[[All, 3]], Apply[UndirectedEdge, dailyedgelistpen[[i, All, 1]], 1],
     EdgeWeight → dailyedgelistpen[[i, All, 2]]]], {i, Length[dailyedgelistpen]}];

Length[distmat]

7633

Length[dailyedgelistpen]

72

dm = Table[distmat[[Table[i, {i, 2 + u, 7632 + u, 72}],
    Table[i, {i, 2 + u, 7632 + u, 72}]]], {u, 0, 71}];

mcs = Table[ppmcc[dm[[order[[i]]]], wms[[order[[i]]]]], {i, 1, 72}];

mcs = Transpose[{dayofyear, mcs}];

reg = LinearModelFit[mcs, {1, x}, x]

FittedModel[ 0.00751738 - 0.000521575 x ]

line = reg["BestFit"]

0.00751738 - 0.000521575 x

cis = reg["MeanPredictionBands"];
```

```
similSNTW = Show[ListLinePlot[mcs, PlotStyle → LightBlue, PlotRange → All],
  ListPlot[mcs], Plot[line, {x, 158, 289}, PlotStyle → Red],
  Plot[cis[[1]], {x, 158, 289}, PlotStyle → {Red, Dashed}],
  Plot[cis[[2]], {x, 158, 289}, PlotStyle → {Red, Dashed}], PlotRange → All,
  ImageSize → 800, Axes → False, Frame → True, FrameStyle → Directive[Thick, Black, 32],
  FrameLabel → {"Matrix correlation coefficient for AI and DTW"},
  LabelStyle → Directive[24, Black]]
```




```
reg["ANOVATable"]
```

	DF	SS	MS	F-Statistic	P-Value
x	1	0.0168844	0.0168844	14.5463	0.000291579
Error	70	0.0812515	0.00116074		
Total	71	0.0981359			

```
reg["AdjustedRSquared"]
```

```
0.160224
```

```
Export["AIxDTWoverTimePen13.tif", similSNTW]
```

```
AIxDTWoverTimePen13.tif
```

Pen 14

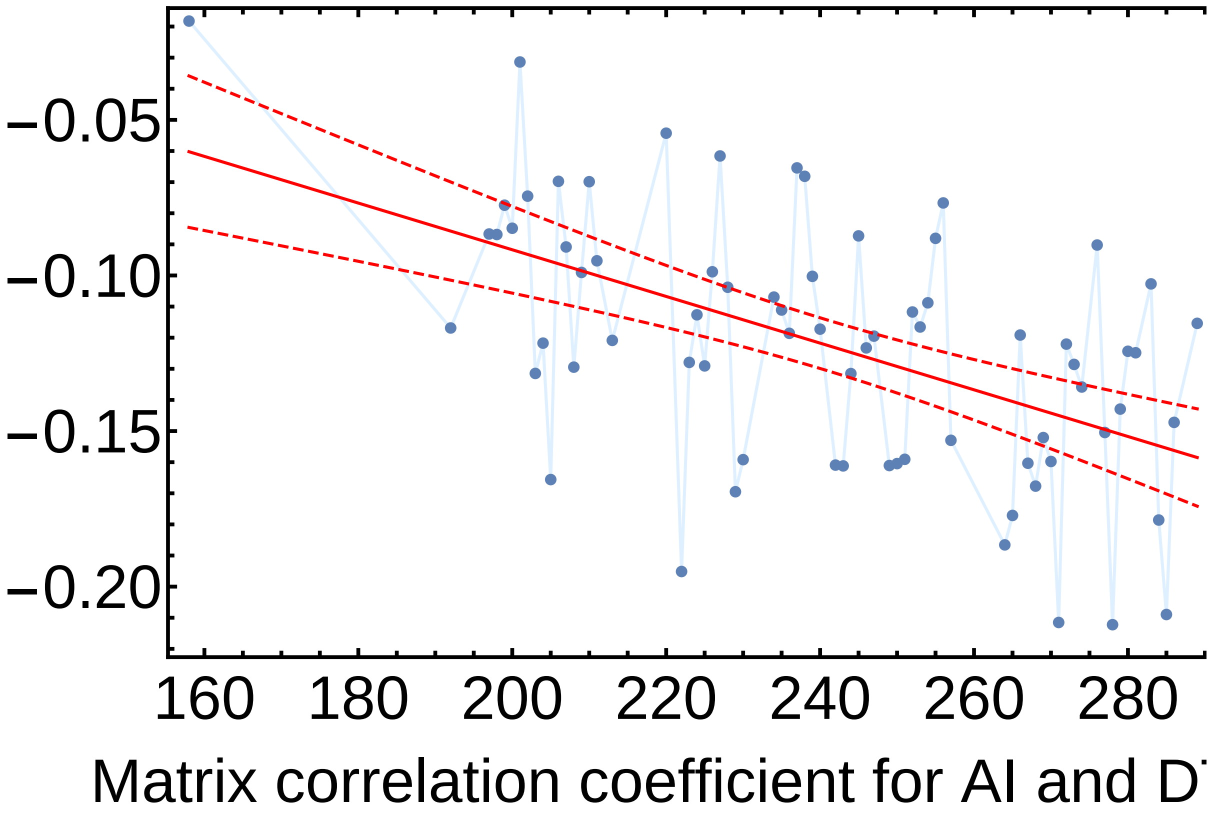
```
distmat = Get["distmat14"];
```

```
hens = Cases[allhens, {_, "14", __}];
```

```
dailyedgelistpen = Get["dailyedgelistpen14"];
```

```
datelist = Get["datelist"]; (*sorted*)

order = {52, 14, 26, 35, 17, 56, 67, 45, 5, 27, 36, 19, 58, 70, 48,
   8, 30, 10, 12, 63, 42, 4, 25, 33, 16, 55, 66, 44, 18, 57, 69, 47, 7, 29,
   38, 60, 72, 39, 1, 22, 31, 51, 62, 41, 3, 24, 32, 15, 54, 65, 68, 46, 6,
   28, 37, 20, 59, 71, 49, 9, 21, 11, 50, 61, 40, 2, 23, 13, 53, 64, 43, 34};
 (*dailyedgelistpen and dtw table is not in chronological order*)

wms = Table[WeightedAdjacencyMatrix[
    Graph[hens〚All, 3〛, Apply[UndirectedEdge, dailyedgelistpen〚i, All, 1〛, 1],
     EdgeWeight → dailyedgelistpen〚i, All, 2〛]], {i, Length[dailyedgelistpen]}];

Length[distmat]

7489

Length[dailyedgelistpen]

72

dm = Table[distmat〚Table[i, {i, 2 + u, 7489 + u, 72}],
    Table[i, {i, 2 + u, 7489 + u, 72}]〛, {u, 0, 71}];

mcs = Table[ppmcc[dm〚order〚i〛〛, wms〚order〚i〛〛], {i, 1, 72}];

mcs = Transpose[{dayofyear, mcs}];

reg = LinearModelFit[mcs, {1, x}, x]

FittedModel[ 0.0582984 - 0.000750173 x ]

line = reg["BestFit"]

0.0582984 - 0.000750173 x

cis = reg["MeanPredictionBands"];
```

```
similSNTW = Show[ListLinePlot[mcs, PlotStyle → LightBlue, PlotRange → All],
  ListPlot[mcs], Plot[line, {x, 158, 289}, PlotStyle → Red],
  Plot[cis[[1]], {x, 158, 289}, PlotStyle → {Red, Dashed}],
  Plot[cis[[2]], {x, 158, 289}, PlotStyle → {Red, Dashed}], PlotRange → All,
  ImageSize → 800, Axes → False, Frame → True, FrameStyle → Directive[Thick, Black, 32],
  FrameLabel → {"Matrix correlation coefficient for AI and DTW"},
  LabelStyle → Directive[24, Black]]
```




```
reg["ANOVATable"]
```

	DF	SS	MS	F-Statistic	P-Value
x	1	0.0349282	0.0349282	29.1398	8.70975×10^{-7}
Error	70	0.083905	0.00119864		
Total	71	0.118833			

```
reg["AdjustedRSquared"]
```

```
0.283839
```

```
Export["AIxDTWoverTimePen14.tif", similSNTW]
```

```
AIxDTWoverTimePen14.tif
```

Combined probability

```
chisquared = -2 (Log[0.027] + Log[0.071] + Log[0.0003] + Log[8.7 × 10^-7])
```

```
56.647
```

```
N[Probability[x >= chisquared, x ≈ ChiSquareDistribution[8]]]
```

2.11011×10^{-9}

Random Numbers

For the time shift test a random list is created. 36 days (half of all days) are randomly selected, on each day 100 dyads are selected for each pen (N=14400 in total). For these dyads DTW distances are re-calculated after shifting one time series by 60 seconds (done in R) and AI are too re-calculated (next section).

```
In[∘]:= SeedRandom[54123]

In[∘]:= daysample = Sort[RandomSample[Range[72], 36]]

Out[∘]= {1, 2, 3, 4, 5, 9, 10, 11, 15, 16, 17, 22, 23, 26, 29, 31, 35, 36,
         37, 39, 41, 43, 46, 50, 51, 54, 55, 59, 60, 61, 63, 65, 66, 70, 71, 72}

In[∘]:= randomlist = Round[Flatten[Table[Table[Table[Join[{daysample⟦d⟧},
              RandomSample[Cases[allhens, {_, ToString[p], __}], 2]⟦All, 1⟧],
            {100}], {d, Length[daysample]}], {p, 11, 14}], 2]];

        Export["randomlist.csv", randomlist]

        randomlist.csv
```

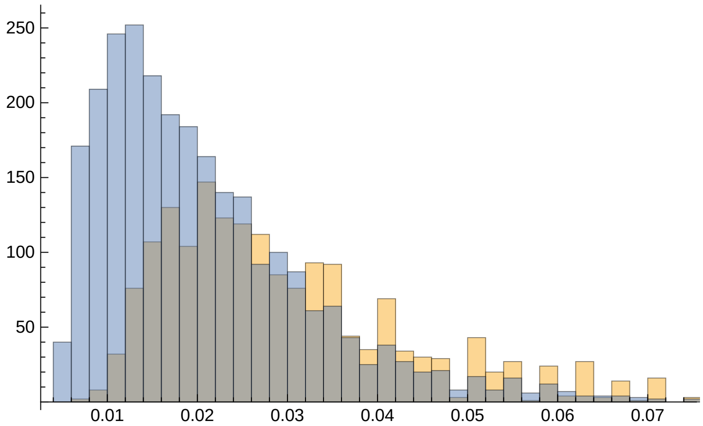
Shift Control

```
In[∘]:= datelist = datelist⟦Ordering[Table[AbsoluteTime[{datelist⟦i⟧,
            {"DayName", " ", "Day", "/", "Month", "/", "Year"}}], {i, Length[datelist]}]]⟧;
```

```
shiftcontrol = {};
For[i = 1, i ≤ Length[randomlist], i++,
 pair = randomlist[[i]];
 pennumber = ToExpression[Cases[allhens, {pair[[2]], __}][[1, 2]]];
 day = datelist[[Round[pair[[1]]]]];
 bird1 = Cases[allhens, {pair[[2]], __}][[1, 3]];
 bird2 = Cases[allhens, {pair[[3]], __}][[1, 3]];
 pen = Cases[data, {_, _, _, pennumber, day, __}];
 pen = Sort[penday, #1[[6]] < #2[[6]] &];
 shift = 0;
 friends = Flatten[ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}], 1];
 ai = (Count[friends, {bird1, bird2}] + Count[friends, {bird2, bird1}]) /
   (Count[friends, {___, bird1, ___}] + Count[friends, {___, bird2, ___}] -
     Count[friends, {bird1, bird2}] - Count[friends, {bird2, bird1}]);
 shift = 60;
 shiftedfriends = Flatten[ParallelTable[findFriend[i], {i, 1, Length[pen] - 1}], 1];
 shiftedai =
  (Count[shiftedfriends, {bird1, bird2}] + Count[shiftedfriends, {bird2, bird1}]) /
   (Count[shiftedfriends, {___, bird1, ___}] +
     Count[shiftedfriends, {___, bird2, ___}] -
     Count[shiftedfriends, {bird1, bird2}] - Count[shiftedfriends, {bird2, bird1}]);
 AppendTo[shiftcontrol, {day, pennumber, bird1, bird2, N[ai], N[shiftedai]}]
]
```

Here AI is calculated for a list of dyads (for given days and pens) twice: once normal and once after shifting the transition times of one bird by 5 seconds. The outcome is a matrix, where each line gives: day, pennumber, birdID1, birdID2, unshifted AI, shifted AI.

```
Histogram[
 {DeleteCases[shiftcontrol[[All, -1]], 0.], DeleteCases[shiftcontrol[[All, -2]], 0.]}]
```



Histogram showing AIs of randomly chosen dyads without and with time shift of 60s.

```
shiftcontrol = shiftcontrol //. Indeterminate → 0.;
```

```
Save["shiftcontrol", shiftcontrol]
```

```
shiftcontrol = Prepend[shiftcontrol,
   {"Date", "Pen", "Individual 1", "Individual 2", "AI", "shifted AI"}];
```

```
Export["shiftcontrol.csv", shiftcontrol]
```

```
shiftcontrol.csv
```

Test with Time Shifted Data

Here the Pearson product moment correlation of the AI of the unshifted and shifted time series and of the DTW distances of unshifted and shifted time series are calculated and graphed.

```
aiunshiftedshifted = Import["shiftcontrol.csv"];
```

```
dtwunshifted = Import["randomlistunshiftedconverted.csv"];
```

```
dtwshifted = Import["randomlist_full_new.csv"]; (*file "randomlist" was re-named*)
```

```
makematrix[line_] := Module[{in, dtws, hex1, hex2, date},
  in = dtwunshifted[[line]];
  dtws = Cases[dtwshifted, {_, in[[1]], in[[2]], in[[3]], _}][[1, 5]];
  hex1 = Cases[allhens, {N[in[[2]]], __}][[1, 3]];
  hex2 = Cases[allhens, {N[in[[3]]], __}][[1, 3]];
  date = datelist[[in[[1]]]];
  Join[in, {dtws}, Cases[aiunshiftedshifted, {date, _, hex1, hex2, __}][[1, 5 ;; 6]]]
 ]
```

```
timeshifttable = Prepend[Table[makematrix[i], {i, 2, Length[dtwunshifted]}],
   {"day", "hen1", "hen2", "dtw", "dtwshifted", "ai", "aishifted"}];
```

```
Save["timeshifttable", timeshifttable]
```

```
tstshort = DeleteCases[timeshifttable, {__, 0., 0., ___}];
```

```
tstshort // Length
```

```
3830
```

```
dtwcorr = Transpose[{Drop[tstshort[[All, 4]], 1], Drop[tstshort[[All, 5]], 1]}];
```

```
N[Correlation[dtwcorr]]
```

```
{{1., 0.999906}, {0.999906, 1.}}
```

Covariance Matrix for DTW

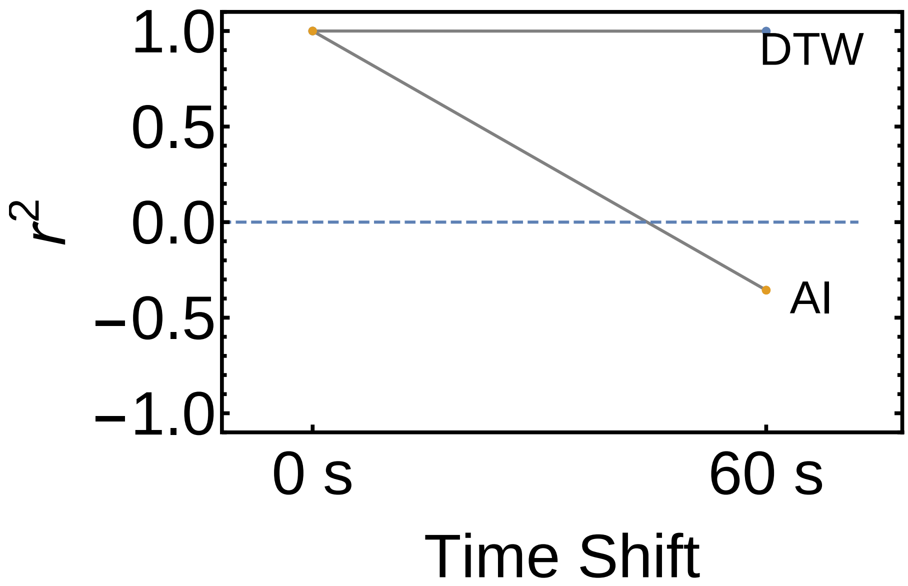
```
aicorr = Transpose[{Drop[tstshort[[All, 6]], 1], Drop[tstshort[[All, 7]], 1]}];
```

```
N[Correlation[aicorr]]
```

```
{{1., -0.355851}, {-0.355851, 1.}}
```

Covariance Matrix for AI

```
scplot = Show[ListLinePlot[{{{1, 1}, {2, 0.999}}, {{1, 1}, {2, -0.356}}},
    PlotRange → {{0.8, 2.3}, {-1.1, 1.1}}, PlotStyle → Gray, PlotRange → All],
   ListPlot[{{{1, 1}, {2, 0.999}}, {{1, 1}, {2, -0.356}}}],
   Plot[0, {x, 0.8, 2.2}, PlotStyle → Dashed],
   Graphics[{Inset[Text[Style["DTW", Large]], {2.1, 0.9}],
     Inset[Text[Style["AI", Large]], {2.1, -0.4}]}], ImageSize → 800, Axes → False,
   Frame → True, FrameTicks → {{True, True}, {{{1, "0 s"}, {2, "60 s"}}, False}},
   FrameStyle → Directive[Thick, Black, 32],
   FrameLabel → {"Time Shift", "r²"}, LabelStyle → Directive[24, Black]]
```




```
Export["shiftcontrolplot.tiff", scplot]
```

shiftcontrolplot.tiff