In[1]:=
```
ParallelNeeds["PairwiseComparisonsNoGIDX`"];
```

In[2]:=
```
<< PairwiseComparisonsNoGIDX`;
```

In[3]:=
```
GetConnectedRandomGraph[noVertex_, noEdges_] :=
  Which[noEdges < (noVertex - 1),
   Null, (* there are no exists connected graph*)
   noEdges == noVertex - 1,
   Graph[Range[noVertex], With[{vertexSeq = RandomSample@Range[noVertex]},
     Map[vertexSeq[[#]] ⟷ vertexSeq[[# + 1]] &, Range[noVertex - 1]]]],
   noEdges > noVertex - 1, With[{randomGraph =
      RandomGraph[{noVertex, noEdges}]},
    If[Length@ConnectedComponents[randomGraph] == 1, randomGraph,
     GetConnectedRandomGraph[noVertex, noEdges]]]];
```

In[4]:=
```
DeteriorateMatrixAccordingToPattern[inputMatrix_, patternMatrix_] :=
  Map[If[#[[2]] == 0, □, #[[1]]] &,
   Transpose /@ Transpose[{inputMatrix, patternMatrix}], {2}];
```

In[5]:=
```
MakeIrreversibleBinaryRandomMatrix[noVertices_, noEdges_] :=
  MapIndexed[If[#2[[1]] == #2[[2]], 1, #1] &, (* set diagonal to 1 *)
   Normal@AdjacencyMatrix@GetConnectedRandomGraph[noVertices, noEdges], {2}];
```

In[6]:=
```
RandomlyIncompleteRandomlyDisturbedMatrix[randMatrix_, noOfComparisons_] :=
  With[{ patternMatrix =
     MakeIrreversibleBinaryRandomMatrix[Length@randMatrix, noOfComparisons]},
   DeteriorateMatrixAccordingToPattern[randMatrix, patternMatrix]];
```

In[7]:=
```
AutoRandomlyIncompleteRandomlyDisturbedMatrix[randMatrix_] :=
  With[{ patternMatrix =
     MakeIrreversibleBinaryRandomMatrix[Length@randMatrix, RandomChoice@
       Range[Length@randMatrix - 1, Binomial[Length@randMatrix, 2]]]},
   DeteriorateMatrixAccordingToPattern[randMatrix, patternMatrix]];
```

In[8]:=
```
AdjacencyMatrixFromPCMatrix[matrix_] :=
  Map[If[NumberQ@#, 1, 0] &, matrix, {2}] - IdentityMatrix[Length@matrix];
```

In[9]:=
```
C_test = RandomlyIncompleteRandomlyDisturbedMatrix[
   RandomMatrix[7, RandomRankingPattern[7, 9], 1.5], 12];
```

In[10]:=
```
C_test // MatrixForm
```

Out[10]//MatrixForm=

1.	0.801778	□	□	1.77973	2.1988	□
1.24723	1.	17.4874	9.47645	□	□	1.45798
□	0.0571841	1.	□	0.0758062	□	□
□	0.105525	□	1.	0.122115	□	0.0789629
0.561883	□	13.1915	8.18901	1.	0.759286	0.621826
0.454794	□	□	□	1.31703	1.	0.803526
□	0.685879	□	12.6642	1.60817	1.24452	1.

In[11]:=
```
MakeIrreversibleBinaryRandomMatrix[7, 18] // MatrixForm
```

Out[11]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

==========

(Generalized) Geometric Consistency Index

==========

In[12]:=
```
GetErrorMatrix[matrix_, w_] :=
  MapIndexed[ #1 * w[[Last@#2]] / w[[First@#2]] &, matrix, {2}];
```

In[13]:=
```
GetErrorMatrix[ {{1, a12, a13, a14, a14}, {a21, 1, a23, a24, a25}, {a31, a32, 1, a34, a35}, {a41, a42, a43, 1, a45}, {a51, a52, a53, a54, 1}}, {w1, w2, w3, w4, w5}] // MatrixForm
```

Out[13]//MatrixForm=

$$\begin{pmatrix} 1 & \frac{a_{12}\, w_2}{w_1} & \frac{a_{13}\, w_3}{w_1} & \frac{a_{14}\, w_4}{w_1} & \frac{a_{14}\, w_5}{w_1} \\ \frac{a_{21}\, w_1}{w_2} & 1 & \frac{a_{23}\, w_3}{w_2} & \frac{a_{24}\, w_4}{w_2} & \frac{a_{25}\, w_5}{w_2} \\ \frac{a_{31}\, w_1}{w_3} & \frac{a_{32}\, w_2}{w_3} & 1 & \frac{a_{34}\, w_4}{w_3} & \frac{a_{35}\, w_5}{w_3} \\ \frac{a_{41}\, w_1}{w_4} & \frac{a_{42}\, w_2}{w_4} & \frac{a_{43}\, w_3}{w_4} & 1 & \frac{a_{45}\, w_5}{w_4} \\ \frac{a_{51}\, w_1}{w_5} & \frac{a_{52}\, w_2}{w_5} & \frac{a_{53}\, w_3}{w_5} & \frac{a_{54}\, w_4}{w_5} & 1 \end{pmatrix}$$

In[14]:=
```
GetGMMErrorMatrix[matrix_] :=
  Map[If[NumberQ@#, #, □] &, GetErrorMatrix[matrix, GMM@matrix], {2}];
```

In[15]:=
```
GetGMMErrorMatrix[Ctest] // MatrixForm
```

Out[15]//MatrixForm=

$$\begin{pmatrix} 1. & 0.777826 & □ & □ & 0.978812 & 1.31346 & □ \\ 1.28563 & 1. & 0.866908 & 0.741377 & □ & □ & 1.21024 \\ □ & 1.15353 & 1. & □ & 0.866908 & □ & □ \\ □ & 1.34884 & □ & 1. & 0.884895 & □ & 0.837814 \\ 1.02165 & □ & 1.15353 & 1.13008 & 1. & 0.824696 & 0.910479 \\ 0.761345 & □ & □ & □ & 1.21257 & 1. & 1.08321 \\ □ & 0.826285 & □ & 1.19358 & 1.09832 & 0.923183 & 1. \end{pmatrix}$$

In[16]:=
```
UpperTriangularizeToUndef[matrix_] :=
  MapIndexed[If[First@#2 ≥ Last@#2, □, #1] &, matrix, {2}];
```

In[17]:=
```
UpperTriangularizeToUndef@ {{1, a12, a13, a14, a14}, {a21, 1, a23, a24, a25}, {a31, a32, 1, a34, a35}, {a41, a42, a43, 1, a45}, {a51, a52, a53, a54, 1}} // MatrixForm
```

Out[17]//MatrixForm=

$$\begin{pmatrix} □ & a_{12} & a_{13} & a_{14} & a_{14} \\ □ & □ & a_{23} & a_{24} & a_{25} \\ □ & □ & □ & a_{34} & a_{35} \\ □ & □ & □ & □ & a_{45} \\ □ & □ & □ & □ & □ \end{pmatrix}$$

Generalized consistency index

In[18]:=
```
(*GCI[matrix_]:=Mean@Map[(Log@#)^2 &,
   Select[Flatten@UpperTriangularizeToUndef@matrix, NumberQ@#&]]*)
```

In[19]:=
```
GCIv1[matrix_] := Mean@Map[(Log@#)^2 &, Select[Flatten@
      UpperTriangularizeToUndef@GetGMMErrorMatrix@matrix, NumberQ@# &]];
```

In[20]:=
```
GCIv2[matrix_] :=
  With[{n = Length@matrix}, 2/((n - 1) (n - 2)) Plus @@ Map[(Log@#)^2 &, Select[Flatten@
        UpperTriangularizeToUndef@GetGMMErrorMatrix@matrix, NumberQ@# &]]];
```

In[21]:=
```
GCIv2[C_test]
```

Out[21]=
```
0.026886
```

==========

(Generalized) Koczkodaj Index

==========

In[22]:=
```
KoczkodajTriadIndex[matrix_, triad_] :=
  With[{i = triad[[1]], k = triad[[2]], j = triad[[3]]},
   With[{a = matrix[[i, k]], b = matrix[[k, j]], c = matrix[[i, j]]},
    Min[Abs[1 - a*b/c], Abs[1 - c/(a*b)]]]];
```

In[23]:=
```
KoczkodajTupleIndex[matrix_, tuple_] :=
  Min[Abs[1 - Product[matrix[[tuple[[i]], tuple[[i + 1]]]],
       {i, 1, Length@tuple - 1}]/matrix[[tuple[[1]], tuple[[-1]]]]],
   Abs[1 - matrix[[tuple[[1]], tuple[[-1]]]]/Product[
       matrix[[tuple[[i]], tuple[[i + 1]]]], {i, 1, Length@tuple - 1}]]];
```

In[24]:=
```
KoczkodajTriadIndex[{{1, a12, a13, a14, a14}, {a21, 1, a23, a24, a25}, {a31, a32, 1, a34, a35}, {a41, a42, a43, 1, a45}, {a51, a52, a53, a54, 1}}, {1, 2, 3}]
```

Out[24]=
$$\text{Min}\left[\text{Abs}\left[1-\frac{a_{13}}{a_{12}\,a_{23}}\right],\ \text{Abs}\left[1-\frac{a_{12}\,a_{23}}{a_{13}}\right]\right]$$

In[25]:=
```
KoczkodajTupleIndex[{{1, a12, a13, a14, a14}, {a21, 1, a23, a24, a25}, {a31, a32, 1, a34, a35}, {a41, a42, a43, 1, a45}, {a51, a52, a53, a54, 1}}, {1, 2, 3, 4}]
```

Out[25]=
$$\text{Min}\left[\text{Abs}\left[1-\frac{a_{14}}{a_{12}\,a_{23}\,a_{34}}\right],\ \text{Abs}\left[1-\frac{a_{12}\,a_{23}\,a_{34}}{a_{14}}\right]\right]$$

In[26]:=
```
TriadExistsQ[matrix_, triad_] :=
  With[{i = triad[[1]], k = triad[[2]], j = triad[[3]]},
   NumberQ@matrix[[i, k]] && NumberQ@matrix[[k, j]] && NumberQ@matrix[[i, j]]];
```

In[27]:=
```
TupleExistsQ[matrix_, tuple_] :=
  And @@ Table[NumberQ@matrix[[tuple[[i]], tuple[[i + 1]]]],
     {i, 1, Length@tuple - 1}] && NumberQ@matrix[[tuple[[1]], tuple[[-1]]]];
```

In[28]:=
```
KoczkodajIndex[matrix_] := KoczkodajIndex[matrix] =
   With[{allTriads = Subsets[Range[Length@matrix], {3}]},
    With[{existingTriads = Select[allTriads, TriadExistsQ[matrix, #] & ]},
     Max@Map[KoczkodajTriadIndex[matrix, #] &, existingTriads]]];
```

In[29]:=
```
(*Unset[GetAllTuplesLongerThan2]*)
```

In[30]:=
```
(*GetAllTuplesLongerThan2[length_]:=(*GetAllTuplesLongerThan2[length] = *)
  Select[Subsets[Range[length]], Length@# ≥ 3 &];*)
```

In[31]:=
```
GetTuplesByCyclesFromAdjacencyMatrix[adjMatrix_] :=
  GetTuplesByCyclesFromAdjacencyMatrix[adjMatrix] =
   Map[First@# &, FindCycle[AdjacencyGraph@adjMatrix, Infinity, All], {2}];
```

In[32]:=
```
SetSharedFunction[GetTuplesByCyclesFromAdjacencyMatrix];
```

In[33]:=
```
GetTuplesByCycles[matrix_] :=
  GetTuplesByCyclesFromAdjacencyMatrix@AdjacencyMatrixFromPCMatrix@matrix;
```

In[34]:=
```
GeneralizedKoczkodajIndex[matrix_] :=
  With[{tuples = GetTuplesByCycles[matrix] },
   If[Length@tuples == 0, 0,
    Max@Map[KoczkodajTupleIndex[matrix, #] &, tuples]]];
```

In[35]:=
```
KoczkodajIndex[C_test]
```

Out[35]=
```
0.486765
```

In[36]:=
```
GeneralizedKoczkodajIndex[C_test]
```

Out[36]=
```
0.679602
```

==========

Triads based average inconsistency indices

==========

In[37]:=
```
TBSI1[matrix_] := With[{allTriads = Subsets[Range[Length@matrix], {3}]},
   With[{existingTriads = Select[allTriads, TriadExistsQ[matrix, #] & ]},
    Mean@Map[KoczkodajTriadIndex[matrix, #] &, existingTriads]]];
```

In[38]:=
```
GeneralizedTBSI1[matrix_] := GeneralizedTBSI1[matrix] =
   With[{tuples = GetTuplesByCycles[matrix] },
    If[Length@tuples == 0, 0,
     Mean@Map[KoczkodajTupleIndex[matrix, #] &, tuples]]];
```

In[39]:=
```
TBSI2[matrix_] := With[{allTriads = Subsets[Range[Length@matrix], {3}]},
   With[{existingTriads = Select[allTriads, TriadExistsQ[matrix, #] & ]},
    With[{trlist = Map[KoczkodajTriadIndex[matrix, #] * KoczkodajTriadIndex[
           matrix, #] &, existingTriads]}, Sqrt[Plus @@ trlist]/Length@trlist ]]];
```

```
In[40]:= GeneralizedTBSI2[matrix_] := GeneralizedTBSI2[matrix] =
           With[{tuples = GetTuplesByCycles[matrix] },
            If[Length@tuples == 0, 0,
             With[{trlist = Map[(KoczkodajTupleIndex[matrix, #])^2 &,  tuples]},
              Sqrt[Plus @@ trlist] / Length@trlist ]]];

In[41]:= (*GetTuplesByCycles@C_test*)

In[42]:= GeneralizedTBSI1[C_test]
Out[42]= 0.359973

In[43]:= GeneralizedTBSI2[C_test]
Out[43]= 0.070866

In[44]:= TBSIAlpha[matrix_, alpha_] :=
           alpha * KoczkodajIndex@matrix + (1 - alpha) * TBSI1@matrix;

In[45]:= GeneralizedTBSIAlpha[matrix_, alpha_] :=
           GeneralizedTBSIAlpha[] = alpha * GeneralizedKoczkodajIndex@matrix +
             (1 - alpha) * GeneralizedTBSI1@matrix;

In[46]:= TBSIAlphaBeta[matrix_, alpha_, beta_] := alpha * KoczkodajIndex@matrix +
            beta * TBSI1@matrix + (1 - alpha - beta) * TBSI2@matrix;

In[47]:= GeneralizedTBSIAlphaBeta[matrix_, alpha_, beta_] :=
           alpha * GeneralizedKoczkodajIndex@matrix + beta * GeneralizedTBSI1@matrix +
            (1 - alpha - beta) * GeneralizedTBSI2@matrix;

In[48]:= GeneralizedTBSIAlpha[C_test, 0.5]
Out[48]= 0.519788

In[49]:= GeneralizedTBSIAlphaBeta[C_test, 0.3, 0.3]
Out[49]= 0.340219
```

==========

Golden-Wang Index

==========

```
In[50]:= CStarMatrix[matrix_] := Transpose@Map[# / Plus @@ # &, Transpose@matrix];

In[51]:= CStarMatrixNoZero[matrix_] := Transpose@Map[# / Plus @@ # &, Transpose@matrix];
```

zamiana niezdefiniowanych na 0

```
In[52]:= CompleteUndefToZero[matrix_] := Map[If[NumberQ@#, #, 0] &, matrix, {2}];
```

In[53]:=
```
CStarMatrix@{{1, a12, a13, a14, a15}, {a21, 1, a23, a24, a25}, {a31, a32, 1, a34, a35}, {a41, a42, a43, 1, a45}, {a51, a52, a53, a54, 1}} // MatrixForm
```

Out[53]//MatrixForm=

$$\begin{pmatrix} \frac{1}{1+a_{21}+a_{31}+a_{41}+a_{51}} & \frac{a_{12}}{1+a_{12}+a_{32}+a_{42}+a_{52}} & \frac{a_{13}}{1+a_{13}+a_{23}+a_{43}+a_{53}} & \frac{a_{14}}{1+a_{14}+a_{24}+a_{34}+a_{54}} & \frac{a_{15}}{1+a_{15}+a_{25}+a_{35}+a_{45}} \\ \frac{a_{21}}{1+a_{21}+a_{31}+a_{41}+a_{51}} & \frac{1}{1+a_{12}+a_{32}+a_{42}+a_{52}} & \frac{a_{23}}{1+a_{13}+a_{23}+a_{43}+a_{53}} & \frac{a_{24}}{1+a_{14}+a_{24}+a_{34}+a_{54}} & \frac{a_{25}}{1+a_{15}+a_{25}+a_{35}+a_{45}} \\ \frac{a_{31}}{1+a_{21}+a_{31}+a_{41}+a_{51}} & \frac{a_{32}}{1+a_{12}+a_{32}+a_{42}+a_{52}} & \frac{1}{1+a_{13}+a_{23}+a_{43}+a_{53}} & \frac{a_{34}}{1+a_{14}+a_{24}+a_{34}+a_{54}} & \frac{a_{35}}{1+a_{15}+a_{25}+a_{35}+a_{45}} \\ \frac{a_{41}}{1+a_{21}+a_{31}+a_{41}+a_{51}} & \frac{a_{42}}{1+a_{12}+a_{32}+a_{42}+a_{52}} & \frac{a_{43}}{1+a_{13}+a_{23}+a_{43}+a_{53}} & \frac{1}{1+a_{14}+a_{24}+a_{34}+a_{54}} & \frac{a_{45}}{1+a_{15}+a_{25}+a_{35}+a_{45}} \\ \frac{a_{51}}{1+a_{21}+a_{31}+a_{41}+a_{51}} & \frac{a_{52}}{1+a_{12}+a_{32}+a_{42}+a_{52}} & \frac{a_{53}}{1+a_{13}+a_{23}+a_{43}+a_{53}} & \frac{a_{54}}{1+a_{14}+a_{24}+a_{34}+a_{54}} & \frac{1}{1+a_{15}+a_{25}+a_{35}+a_{45}} \end{pmatrix}$$

In[54]:=
```
GMMMatrix[matrix_] :=
  With[{g = N@GMM@matrix}, Transpose@Table[g, {x, 1, Length@matrix}]];
```

In[55]:=
```
DiffMatrix[matrix_] :=
  CStarMatrix@CompleteUndefToZero@DeteriorateMatrixAccordingToPattern[
      GMMMatrix[matrix], CompleteUndefToZero@matrix] -
   CStarMatrix@CompleteUndefToZero@matrix;
```

W niektórych miejscach jednak zero zostało odjęte od elementów nieokreślonych i trzeba będzie to usunąć.

In[56]:=
```
PreprocessedDiffMatrix[matrix_] := DeteriorateMatrixAccordingToPattern[
   Abs@DiffMatrix[matrix], CompleteUndefToZero@matrix];
```

In[57]:=
```
CStarMatrix@CompleteUndefToZero@
   DeteriorateMatrixAccordingToPattern[N@GMMMatrix@{{1, 2, 3, □}, {1/2, 1, 3, 4}, {1/3, 1/3, 1, 5}, {□, 1/4, 1/5, 1}},
    CompleteUndefToZero@{{1, 2, 3, □}, {1/2, 1, 3, 4}, {1/3, 1/3, 1, 5}, {□, 1/4, 1/5, 1}}] // MatrixForm
```

Out[57]//MatrixForm=

$$\begin{pmatrix} 0.5397 & 0.514358 & 0.514358 & 0. \\ 0.296657 & 0.282728 & 0.282728 & 0.582173 \\ 0.163643 & 0.15596 & 0.15596 & 0.321142 \\ 0. & 0.0469543 & 0.0469543 & 0.0966851 \end{pmatrix}$$

In[58]:=
```
N@CStarMatrix@CompleteUndefToZero@{{1, 2, 3, □}, {1/2, 1, 3, 4}, {1/3, 1/3, 1, 5}, {□, 1/4, 1/5, 1}} // MatrixForm
```

Out[58]//MatrixForm=

$$\begin{pmatrix} 0.545455 & 0.55814 & 0.416667 & 0. \\ 0.272727 & 0.27907 & 0.416667 & 0.4 \\ 0.181818 & 0.0930233 & 0.138889 & 0.5 \\ 0. & 0.0697674 & 0.0277778 & 0.1 \end{pmatrix}$$

In[59]:=
```
DiffMatrix@{{1, 2, 3, □}, {1/2, 1, 3, 4}, {1/3, 1/3, 1, 5}, {□, 1/4, 1/5, 1}} // MatrixForm
```

Out[59]//MatrixForm=

-0.00575489	-0.0437811	0.0976918	0.
0.0239296	0.00365784	-0.133939	0.182173
-0.0181748	0.0629364	0.0170708	-0.178858
0.	-0.0228132	0.0191765	-0.00331493

In[60]:=
```
PreprocessedDiffMatrix@{{1, 2, 3, □}, {1/2, 1, 3, 4}, {1/3, 1/3, 1, 5}, {□, 1/4, 1/5, 1}} // MatrixForm
```

Out[60]//MatrixForm=

0.00575489	0.0437811	0.0976918	□
0.0239296	0.00365784	0.133939	0.182173
0.0181748	0.0629364	0.0170708	0.178858
□	0.0228132	0.0191765	0.00331493

In[61]:=
```
GGWI[matrix_] := 1 / Length@matrix *
    Plus @@ Select[Flatten@PreprocessedDiffMatrix@matrix, NumberQ];
```

In[62]:=
```
GGWI@{{1, 2, 3, □}, {1/2, 1, 3, 4}, {1/3, 1/3, 1, 5}, {□, 1/4, 1/5, 1}}
```

Out[62]= 0.203318

In[63]:=
```
GGWI2[matrix_] := Mean@Select[Flatten@PreprocessedDiffMatrix@matrix, NumberQ];
```

In[64]:=
```
GGWI@C_test
```

Out[64]= 0.134868

In[65]:=
```
GGWI2@C_test
```

Out[65]= 0.030454

==========

Salo-Hamalainen Index

==========

GetTuplesByPath - Find all paths between the given two vertices in a graph

In[66]:=
```
GetTuplesByPath[adjmatrix_, i_, j_] := GetTuplesByPath[adjmatrix, i, j] =
    FindPath[AdjacencyGraph@adjmatrix, i, j, Infinity, All];
```

In[67]:=
```
GetProductByTuple[matrix_, tuple_] :=
  Times @@ Table[matrix[[tuple[[i]], tuple[[i + 1]]]], {i, 1, Length@tuple - 1}];
```

In[68]:=
```
GetEstimationList[matrix_, i_, j_] := Map[GetProductByTuple[matrix, #] &,
    GetTuplesByPath[AdjacencyMatrixFromPCMatrix@matrix, i, j]];
```

In[69]:=
```
RMatrix[matrix_] := Table[With[{estList = GetEstimationList[matrix, i, j]},
    {Min@estList, Max@estList}], {i, 1, Length@matrix}, {j, 1, Length@matrix}];
```

In[70]:=
```
GSHI[matrix_] := With[{n = Length@matrix, rm = RMatrix[matrix]}, 2/(n (n - 1)) Sum[Sum[
    (Last@rm[[i, j]] - First@rm[[i, j]])/((1 + Last@rm[[i, j]]) (1 + First@rm[[i, j]])), {j, i + 1, n}], {i, 1, n - 1}]];
```

In[71]:=
```
C_test // MatrixForm
```

Out[71]//MatrixForm=

$$\begin{pmatrix} 1. & 0.801778 & \square & \square & 1.77973 & 2.1988 & \square \\ 1.24723 & 1. & 17.4874 & 9.47645 & \square & \square & 1.45798 \\ \square & 0.0571841 & 1. & \square & 0.0758062 & \square & \square \\ \square & 0.105525 & \square & 1. & 0.122115 & \square & 0.0789629 \\ 0.561883 & \square & 13.1915 & 8.18901 & 1. & 0.759286 & 0.621826 \\ 0.454794 & \square & \square & \square & 1.31703 & 1. & 0.803526 \\ \square & 0.685879 & \square & 12.6642 & 1.60817 & 1.24452 & 1. \end{pmatrix}$$

In[72]:=
```
RMatrix@C_test // MatrixForm
```

Out[72]//MatrixForm=

$$\begin{pmatrix} \begin{pmatrix} \infty \\ -\infty \end{pmatrix} & \begin{pmatrix} 0.744744 \\ 2.50245 \end{pmatrix} & \begin{pmatrix} 12.2395 \\ 43.7612 \end{pmatrix} & \begin{pmatrix} 7.05754 \\ 40.3348 \end{pmatrix} & \begin{pmatrix} 0.927831 \\ 3.13 \end{pmatrix} & \begin{pmatrix} 0.704489 \\ 2.79057 \end{pmatrix} & \begin{pmatrix} \\ \end{pmatrix} \\ \begin{pmatrix} 0.399609 \\ 1.34274 \end{pmatrix} & \begin{pmatrix} \infty \\ -\infty \end{pmatrix} & \begin{pmatrix} 9.94339 \\ 47.6454 \end{pmatrix} & \begin{pmatrix} 9.47645 \\ 29.5772 \end{pmatrix} & \begin{pmatrix} 0.75377 \\ 3.61181 \end{pmatrix} & \begin{pmatrix} 0.878659 \\ 2.89677 \end{pmatrix} & \begin{pmatrix} \\ \end{pmatrix} \\ \begin{pmatrix} 0.0228513 \\ 0.0817026 \end{pmatrix} & \begin{pmatrix} 0.0209884 \\ 0.100569 \end{pmatrix} & \begin{pmatrix} \infty \\ -\infty \end{pmatrix} & \begin{pmatrix} 0.198896 \\ 1.69134 \end{pmatrix} & \begin{pmatrix} 0.0431037 \\ 0.206538 \end{pmatrix} & \begin{pmatrix} 0.0318034 \\ 0.179647 \end{pmatrix} & \begin{pmatrix} 0 \\ \end{pmatrix} \\ \begin{pmatrix} 0.0247925 \\ 0.141693 \end{pmatrix} & \begin{pmatrix} 0.0338099 \\ 0.105525 \end{pmatrix} & \begin{pmatrix} 0.591246 \\ 5.02776 \end{pmatrix} & \begin{pmatrix} \infty \\ -\infty \end{pmatrix} & \begin{pmatrix} 0.0475031 \\ 0.381135 \end{pmatrix} & \begin{pmatrix} 0.0545137 \\ 0.305681 \end{pmatrix} & \begin{pmatrix} 0 \\ \end{pmatrix} \\ \begin{pmatrix} 0.319488 \\ 1.07778 \end{pmatrix} & \begin{pmatrix} 0.276869 \\ 1.32666 \end{pmatrix} & \begin{pmatrix} 4.84172 \\ 23.1999 \end{pmatrix} & \begin{pmatrix} 2.62374 \\ 21.0513 \end{pmatrix} & \begin{pmatrix} \infty \\ -\infty \end{pmatrix} & \begin{pmatrix} 0.419536 \\ 2.36983 \end{pmatrix} & \begin{pmatrix} \\ \end{pmatrix} \\ \begin{pmatrix} 0.35835 \\ 1.41947 \end{pmatrix} & \begin{pmatrix} 0.345212 \\ 1.1381 \end{pmatrix} & \begin{pmatrix} 5.56646 \\ 31.4432 \end{pmatrix} & \begin{pmatrix} 3.27139 \\ 18.344 \end{pmatrix} & \begin{pmatrix} 0.421972 \\ 2.38359 \end{pmatrix} & \begin{pmatrix} \infty \\ -\infty \end{pmatrix} & \begin{pmatrix} \\ \end{pmatrix} \\ \begin{pmatrix} 0.274083 \\ 1.76655 \end{pmatrix} & \begin{pmatrix} 0.428174 \\ 1.41638 \end{pmatrix} & \begin{pmatrix} 6.92755 \\ 63.6725 \end{pmatrix} & \begin{pmatrix} 4.21941 \\ 20.2864 \end{pmatrix} & \begin{pmatrix} 0.525151 \\ 4.82677 \end{pmatrix} & \begin{pmatrix} 0.602653 \\ 3.81107 \end{pmatrix} & \end{pmatrix}$$

In[73]:=
```
GSHI@C_test
```

Out[73]= 0.247745

==========

Relative Error (Barzilai) Index

==========

In[74]:=
```
XMatrix[matrix_] := With[{delta = Mean@Transpose@matrix, n = Length@matrix},
   Table[delta[[i]] - delta[[j]], {i, 1, n}, {j, 1, n}]];
```

In[75]:=
```
EMatrix[matrix_] := matrix - XMatrix[matrix];
```

In[76]:=
```
REI[matrix_] := Plus @@ ((Flatten@EMatrix[matrix])^2) / Plus @@ ((Flatten@matrix)^2);
```

In[77]:=
```
RCI[matrix_] := Plus @@ ((Flatten@XMatrix[matrix])^2) / Plus @@ ((Flatten@matrix)^2);
```

In[78]:=
```
matrix = {{1, 1, 2, 3}, {1, 1, 4, 5}, {1/2, 1/4, 1, 6}, {1/3, 1/5, 1/6, 1}};
```

```
In[79]:= GetProperEntry[i_, j_, matrix_, gmm_] :=
           If[NumberQ@matrix[[i, j]], matrix[[i, j]], gmm[[i]] / gmm[[j]]];

In[80]:= GetProperEntryZero[i_, j_, matrix_, gmm_] :=
           If[NumberQ@matrix[[i, j]], matrix[[i, j]], 1];

In[81]:= REv1[matrix_] := With[{gmm = GMM[matrix], n = Length@matrix},
           Sum[(Log[GetProperEntry[i, j, matrix, gmm]] - Log[gmm[[i]] / gmm[[j]]])^2,
             {i, 1, n}, {j, 1, n}] /
            Sum[Log[GetProperEntry[i, j, matrix, gmm]]^2, {i, 1, n}, {j, 1, n}]];

In[82]:= (*RC[matrix_]:=With[{gmm = GMM[matrix], n=Length@matrix},
           Sum[(Log[gmm[[i]]/gmm[[j]]])^2, {i,1,n}, {j,1,n}]
         Sum[Log[GetProperEntry[i,j,matrix,gmm]]^2,{i,1,n}, {j,1,n}]];*)

In[83]:= REv2[matrix_] := With[{gmm = GMM[matrix], n = Length@matrix},
           Sum[(Log[GetProperEntry[i, j, matrix, gmm]] - Log[gmm[[i]] / gmm[[j]]])^2,
             {i, 1, n}, {j, 1, n}] /
            Sum[Log[GetProperEntryZero[i, j, matrix, gmm]]^2, {i, 1, n}, {j, 1, n}]];

In[84]:= (*N@RC[matrix] + N@RE[matrix]*)

In[85]:= (*{N@REI@Log@matrix, N@RE@matrix}*)

In[86]:= matrixIncompl = ( 1   1   □   3
                           1   1   □   5
                           □   □   1   6
                           1/3 1/5 1/6 1 );

In[87]:= N@REv1@matrixIncompl
Out[87]= 0.0117457

In[88]:= N@REv2@matrixIncompl
Out[88]= 0.0124123
```

==========

Saaty - Harker Index

==========

```
In[89]:= SaatyIndex[matrix_] := SaatyIdx@HarkerMatrix@matrix;

In[90]:= SaatyIndex@C_test
Out[90]= 0.0096764
```

==========

LLS index (based on GMM 4 incomplete PCM)

==========

```
In[91]:= LLSIndexOriginal[matrix_] := With[{llsm = LLSM@matrix}, Plus @@ Flatten@
            MapIndexed[With[{res = (#1 - llsm[[ #2[[1]] ]] / llsm[[ #2[[2]] ]])^2},
               If[NumberQ@res, res, 0] ] & , matrix, {2}]];

In[92]:= Timing@LLSIndexOriginal@C_test
Out[92]= {0.014781, 26.957}
```

In[93]:=
```
LLSIndex[matrix_] := With[{llsm = GMM@matrix}, Plus @@ Flatten@
      MapIndexed[With[{res = (#1 - llsm[[ #2[[1]] ]] / llsm[[ #2[[2]] ]])^2},
         If[NumberQ@res, res, 0] ] & , matrix, {2}]];
```

In[94]:=
```
Timing@LLSIndex@C_test
```

Out[94]=
```
{0.000857, 26.957}
```

==========

Oliva et al. index

==========

In[95]:=
```
HadamardMatrixProd[matrix1_, matrix2_] :=
  Map[Times @@ # &, Transpose /@ Transpose@{matrix1, matrix2}, {2}];
```

In[96]:=
```
HarkerMatrixPlaceHolderCount[matrix_, row_] :=
  Length[Select[matrix[[row]], Not[And[NumberQ[#], # > 0]] &]];
```

In[97]:=
```
HarkerMatrix[matrix_] := MapIndexed[
    If[#2[[1]] == #2[[2]], HarkerMatrixPlaceHolderCount[matrix, #2[[1]]],
If[And[NumberQ[#1], #1 > 0] , #1, 0]] & , matrix, {2}] +
   IdentityMatrix[First[Dimensions[matrix]]];
```

In[98]:=
```
DMatrix[matrix_] := With[{n = Length@matrix}, Table[If[i == j,
     n - 1 - HarkerMatrixPlaceHolderCount[matrix, i], 0], {i, 1, n}, {j, 1, n}]];
```

In[99]:=
```
Zeroes[matrix_] := Map[If[NumberQ@#, #, 0] &, matrix, {2}];
```

In[100]:=
```
SMatrix[matrix_] := Zeroes[matrix] - IdentityMatrix[Length@matrix];
```

In[101]:=
```
OlivaSetolaScalaIdx[matrix_] :=
  N@Chop@First@Reverse@Map[Abs@# &, SortBy[Eigenvalues[
          Inverse@DMatrix[matrix].SMatrix[matrix]], Abs]] - 1;
```

In[102]:=
```
SMatrix@C_test // MatrixForm
```

Out[102]//MatrixForm=

$$\begin{pmatrix} 0. & 0.801778 & 0 & 0 & 1.77973 & 2.1988 & 0 \\ 1.24723 & 0. & 17.4874 & 9.47645 & 0 & 0 & 1.45798 \\ 0 & 0.0571841 & 0. & 0 & 0.0758062 & 0 & 0 \\ 0 & 0.105525 & 0 & 0. & 0.122115 & 0 & 0.0789629 \\ 0.561883 & 0 & 13.1915 & 8.18901 & 0. & 0.759286 & 0.621826 \\ 0.454794 & 0 & 0 & 0 & 1.31703 & 0. & 0.803526 \\ 0 & 0.685879 & 0 & 12.6642 & 1.60817 & 1.24452 & 0. \end{pmatrix}$$

In[103]:=
```
OlivaSetolaScalaIdx@C_test
```

Out[103]=
```
0.0169275
```

==========

Montecarlo Part

==========

In[111]:=
```
Abort[]
```

Out[111]=
```
$Aborted
```

In[112]:=
```
SetDirectory[NotebookDirectory[]];
```

In[113]:=
```
noOfAlternatives = 7;
```

In[114]:=
```
scaleSize = 9;
```

In[115]:=
```
noOfRankingPatterns = 1000;
```

In[116]:=
```
CloseKernels[];
```

In[117]:=
```
LaunchKernels[7];
```

In[118]:=
```
rankingPatterns = ParallelTable[
    RandomRankingPattern[noOfAlternatives, 9], noOfRankingPatterns];
```

In[120]:=
```
RandomlyIncompleteRandomlyDisturbedMatrix[
  RandomRationalMatrix[noOfAlternatives, scaleSize,
   RandomRationalUniformRankingPattern[noOfAlternatives, scaleSize],
   1], 12] // MatrixForm
```

Out[120]//MatrixForm=

$$\begin{pmatrix} 1 & \square & \frac{3}{2} & \square & \frac{1}{3} & \frac{1}{9} & \square \\ \square & 1 & \square & \square & \frac{1}{2} & \frac{1}{9} & \frac{1}{9} \\ \frac{2}{3} & \square & 1 & \square & \frac{2}{9} & \square & \frac{1}{9} \\ \square & \square & \square & 1 & 6 & \frac{3}{4} & \square \\ 3 & 2 & \frac{9}{2} & \frac{1}{6} & 1 & \square & \frac{1}{9} \\ 9 & 9 & \square & \frac{4}{3} & \square & 1 & \frac{1}{2} \\ \square & 9 & 9 & \square & 9 & 2 & 1 \end{pmatrix}$$

In[121]:=
```
dataTriads = {};
```

In[122]:=
```
randomDataFile = "randomMatrices_v4.mx";
```

In[123]:=
```
FileExistsQ[randomDataFile]
```

Out[123]=
```
True
```

In[124]:=
```
randomCompleteMatrices = {};
```

In[125]:=
```
randomAllMatrices = {};
```

In[126]:=
```
If[FileExistsQ[randomDataFile],
  Get[randomDataFile],
  randomCompleteMatrices =
   ParallelTable[RandomMatrix[noOfAlternatives,
     rankingPatterns[[patNo]], disturbanceLevel] ,
    {disturbanceLevel, 1, 30, 1}, {patNo, 1, 1000}]; randomAllMatrices =
   ParallelMap[Table[ RandomlyIncompleteRandomlyDisturbedMatrix[#,
       noOfComparisons], {noOfComparisons, Binomial[noOfAlternatives, 2],
       noOfAlternatives - 1, -1}] &, randomCompleteMatrices, {2}];
  DumpSave[randomDataFile, randomAllMatrices]];
```

In[127]:=
```
recordsProcessed = 0;
```

In[128]:=
```
SetSharedVariable[recordsProcessed];
```

In[129]:=
```
(*Abort[]*)
```

In[130]:=
```
ComputeIndices[matrix_] := {
   GCIv2@matrix,  (*geometric consistency index v1 *)
   GCIv1@matrix, (*geometric consistency index v2 *)
   GeneralizedKoczkodajIndex@matrix,
   GeneralizedTBSI1@matrix,
   GeneralizedTBSI2@matrix,
   GeneralizedTBSIAlpha[matrix, 0.5],
   GeneralizedTBSIAlphaBeta[matrix, 0.3, 0.3],
   GGWI@matrix, (* Golden-Wang index *)
   GSHI@matrix, (* Salo-Hamalainen index *)
   REv1@matrix, (* Barzilai's error v1 *)
   REv2@matrix, (* Barzilai's error v1 *)
   SaatyIndex@matrix,
   LLSIndex@matrix, (* logarithmic least square criterion *)
   OlivaSetolaScalaIdx@matrix (* oliva et al. index *)
  };
```

In[131]:=
```
ComputeIndices[C_test]
```

Out[131]=
```
{0.026886, 0.0336076, 0.679602, 0.359973, 0.070866, 0.519788, 0.340219,
 0.134868, 0.247745, 0.00597691, 0.0125013, 0.0096764, 26.957, 0.0169275}
```

In[132]:=
```
ComputeIndices[randomAllMatrices[[30, 100, 12]]]
```

Out[132]=
```
{1.6544, 2.4816, 0.999595, 0.923926, 0.294161, 0.961761, 0.694721,
 0.783808, 0.857427, 0.456041, 0.635435, 0.924392, 2865.84, 1.77076}
```

In[134]:=
```
ComputeDataEntity[matrix_] := With[{}, recordsProcessed++;
   ComputeIndices[matrix]];
```

In[135]:=
```
allRecordsInChunk =
 Length@randomAllMatrices[[1]] * Length@randomAllMatrices[[1, 1]]
```

Out[135]= 16 000

Uklad indeksow: [disturbance level, pattern #no, missing comparisons]

In[136]:=
```
randomAllMatrices[[30, 1000, 16]] // MatrixForm
```

Out[136]//MatrixForm=

1.	16.4719	□	1.77288	□	□	□
0.0607093	1.	□	□	0.700865	□	□
□	□	1.	□	0.0237559	□	□
0.564054	□	□	1.	□	15.835	□
□	1.42681	42.0947	□	1.	□	□
□	□	□	0.0631514	□	1.	32.6251
□	□	□	□	□	0.0306513	1.

In[140]:=
```
startTime = 0;
```

In[141]:=
```
endTime = 0;
```

In[142]:=
```
chunkId = 1;
```

In[143]:=
```
{ProgressIndicator[Dynamic[recordsProcessed], {0, allRecordsInChunk}],
 "record processed in chunk:",
 Dynamic[recordsProcessed], "chunkId: ", Dynamic[chunkId]}
```

Out[143]= { , record processed in chunk:, 0, chunkId: , 31}

In[144]:=
```
dataChunk = {};
```

In[145]:=
```
resultsFilePrefix = "results_v4";
```

In[146]:=
```
computeResults = {};
```

In[147]:=
```
ResultsFileName[chunkId_] :=
  resultsFilePrefix <> ".part" <> ToString[chunkId] <> ".mx";
```

In[148]:=
```
FileExistsQ[ResultsFileName[1]]
```

Out[148]= True

In[149]:=
```
For[chunkId = 1, chunkId ≤ Length@randomAllMatrices, chunkId++,
 recordsProcessed = 0;
 If[FileExistsQ[ResultsFileName[chunkId]],
  Print["chunk " <> ToString@chunkId <> " skipped"],
  dataChunk = randomAllMatrices[[chunkId]];
  startTime = AbsoluteTime[];
  computeResults = ParallelMap[ComputeDataEntity[#] &, dataChunk, {2}];
  DumpSave[ResultsFileName[chunkId], computeResults];
  endTime = AbsoluteTime[];
  Print[(endTime - startTime)]];
]
```

```
chunk 1 skipped
chunk 2 skipped
chunk 3 skipped
chunk 4 skipped
chunk 5 skipped
chunk 6 skipped
chunk 7 skipped
chunk 8 skipped
chunk 9 skipped
chunk 10 skipped
chunk 11 skipped
chunk 12 skipped
chunk 13 skipped
chunk 14 skipped
chunk 15 skipped
chunk 16 skipped
chunk 17 skipped
chunk 18 skipped
chunk 19 skipped
chunk 20 skipped
chunk 21 skipped
chunk 22 skipped
chunk 23 skipped
chunk 24 skipped
chunk 25 skipped
chunk 26 skipped
chunk 27 skipped
chunk 28 skipped
chunk 29 skipped
chunk 30 skipped
```

In[150]:=
```
computeResults = {};
```

In[151]:=
```
computeResultsTotal = {};
```

In[152]:=
```
For[i = 1, i ≤ Length@randomAllMatrices, i++,
  Get[ResultsFileName[i]];
  computeResultsTotal = Append[computeResultsTotal, computeResults];
  computeResults = {}];
```

In[153]:=
```
Length@computeResultsTotal[[1, 1, 1]]
```

Out[153]= 14

Correctness check

In[154]:=
```
computeResultsTotal[[20, 1000, 14]] -
 ComputeIndices[randomAllMatrices[[20, 1000, 14]]]
```

Out[154]=
```
{0.146224, -0.146224, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

In[155]:=
```
With[{x1 = RandomInteger[{1, 30}], x2 = RandomInteger[{1, 1000}],
  x3 = RandomInteger[{1, 16}]  }, {computeResultsTotal[[x1, x2, x3]] -
   ComputeIndices[randomAllMatrices[[x1, x2, x3]]], computeResultsTotal[[
   x1, x2, x3]], ComputeIndices[randomAllMatrices[[x1, x2, x3]]]}]
```

Out[155]=
```
{{0.157731, -0.157731, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.},
 {0.337996, 0.180264, 0.928447, 0.745363, 0.448894, 0.836905, 0.6817,
  0.292034, 0.253176, 0.00946465, 0.0731717, 0.0886632, 52.7433, 0.204411},
 {0.180264, 0.337996, 0.928447, 0.745363, 0.448894, 0.836905, 0.6817,
  0.292034, 0.253176, 0.00946465, 0.0731717, 0.0886632, 52.7433, 0.204411}}
```

In[157]:=
```
DistanceMeasure[idxCompl_, idxIncompl_] := If[idxCompl + idxIncompl == 0,
   0, (idxCompl - idxIncompl) / (idxCompl + idxIncompl)];
```

In[158]:=
```
DistanceMeasure2[idxCompl_, idxIncompl_] :=
  If[idxCompl ≤ 10^-7 && idxIncompl ≤ 10^-7, 0,
   (idxCompl - idxIncompl) / Max[idxCompl, idxIncompl]];
```

Distances between complete PC matrices and their incomplete counterparts

In[168]:=
```
distancesArray2 = Table[
   Table[
    With[{completeMatrix = computeResultsTotal[[
        disturbanceLevel, matrixNumber, 1]], incompleteMatrix =
       computeResultsTotal[[disturbanceLevel, matrixNumber, missingComp]]},
     MapThread[DistanceMeasure2, {completeMatrix, incompleteMatrix}]
    ], {missingComp, 1, 16}],
   {disturbanceLevel, 1, 30}, {matrixNumber, 1, 1000}];
```

In[173]:=
```
OlivaSetolaScalaIdx@randomAllMatrices[[30, 4, 16]]
```

Out[173]=
$$1.55431 \times 10^{-15}$$

In[176]:=
```
distancesArrayFlatten2 = Flatten[distancesArray2, 1];
```

In[177]:=
```
Length@distancesArrayFlatten2
```

Out[177]=
```
30 000
```

In[178]:=
```
distancesArrayFlatten2[[1000, 15, 13]]
```

Out[178]=
```
0
```

In[180]:=
```
GetMeanForIndexNoAndMissingComp2[idxNo_, missingComp_] :=
  Mean@Table[distancesArrayFlatten2[[x, missingComp + 1, idxNo]],
    {x, 1, Length@distancesArrayFlatten2}];
```

In[181]:=
```
GetMeanForIndexNoAndMissingComp2[14, 15]
```

Out[181]=
```
0.966667
```

In[182]:=
```
GetMeanForIndexNoAndMissingComp2tmp[idxNo_, missingComp_] :=
  Table[distancesArrayFlatten2[[x, missingComp + 1, idxNo]],
   {x, 1, Length@distancesArrayFlatten2}];
```

In[186]:=
```
indxNames =
  {"Geometric Consistency index v2", "Geometric Consistency index v1",
   "Koczkodaj's index", "Triad based index 1", "Triad based index 2",
   "Triad based alpha index", "Triad based alpha-beta index",
   "Golden-Wang index", "Salo-Hamalainen index",
   "Barzilai's Relative Error index v1", "Barzilai's Relative Error index v2",
   "Saaty consistency index", "iLLS criterion", "Oliva et al. index"};
```

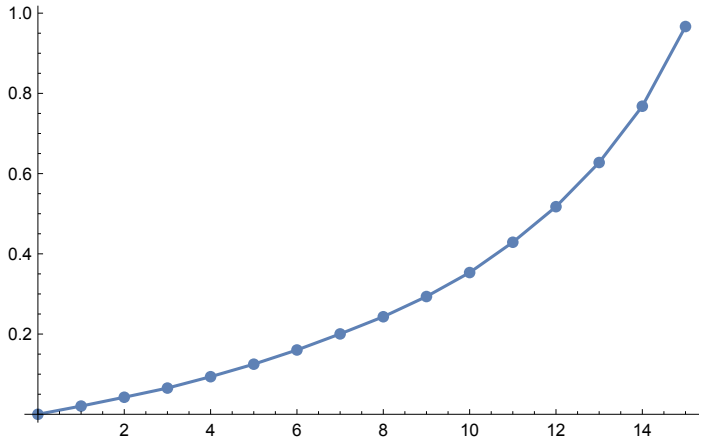
In[187]:=
```
Length@indxNames
```

Out[187]= 14

In[215]:=
```
List2DPlotByIdxNo2[idxNo_] := List2DPlotByIdxNo2[idxNo] =
   Table[{missingComp, GetMeanForIndexNoAndMissingComp2[idxNo, missingComp]},
    {missingComp, 0, 15}];
```

In[216]:=
```
ListPlot[List2DPlotByIdxNo2[14], Joined → True,  PlotMarkers → Automatic]
```

Out[216]=



In[196]:=

In[197]:=
```
(*Abort[]*)
```

In[198]:=
```
Length@indxNames
```

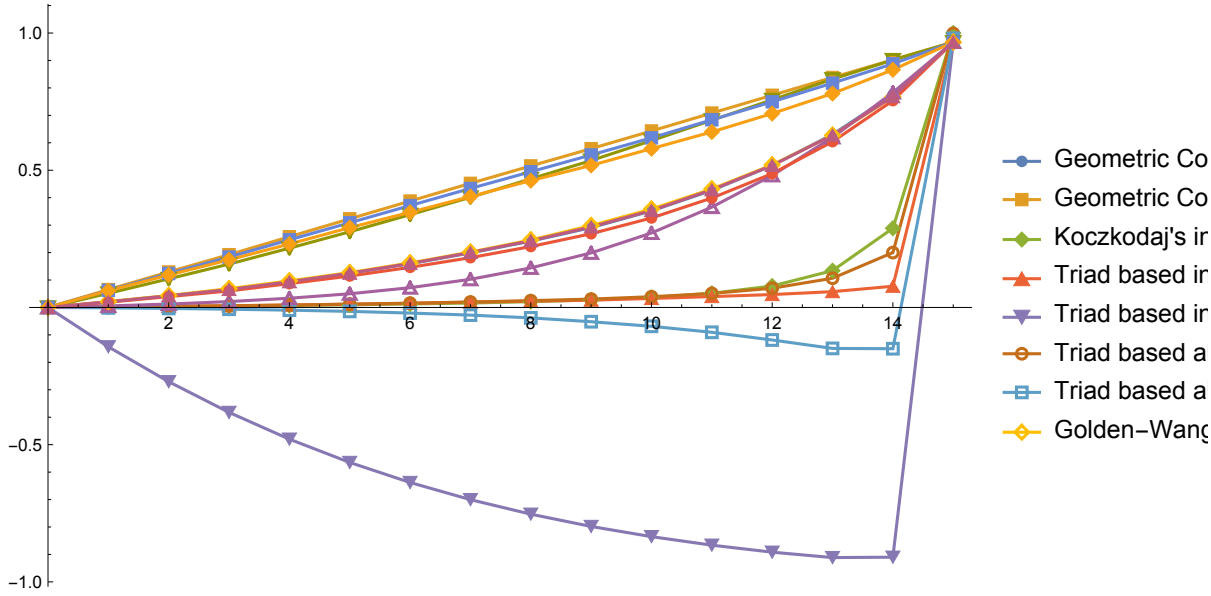
Out[198]= 14

In[199]:=
```
GetMeanForIndexNoAndMissingComp2[14, 11]
```

Out[199]= 0.428892

In[202]:=
```
ListPlot[
 ParallelTable[List2DPlotByIdxNo2[idxNo], {idxNo, 1, Length@indxNames}],
 Joined → True,  PlotMarkers → Automatic, PlotLegends → indxNames]
```

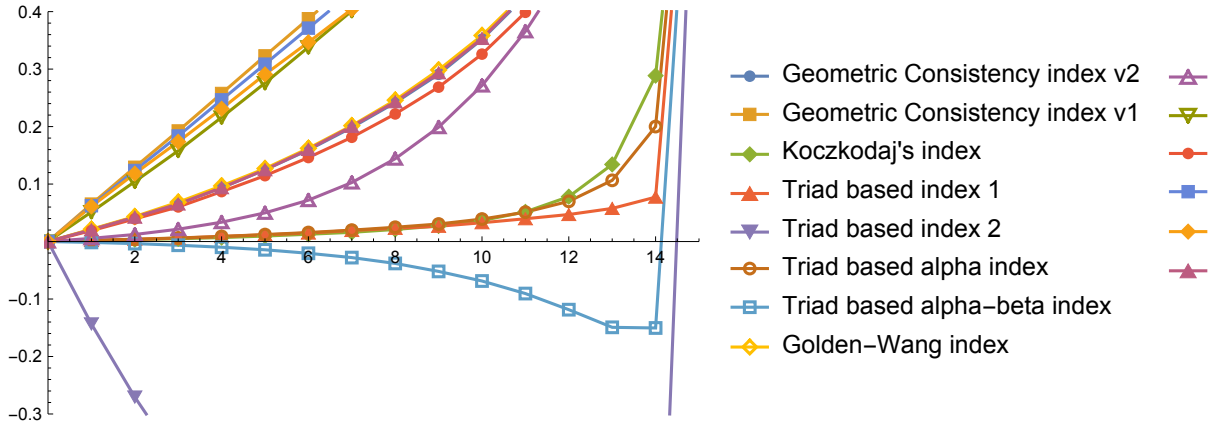
Out[202]=



In[203]:=

In[204]:=
```
ListPlot[ParallelTable[List2DPlotByIdxNo2[idxNo],
  {idxNo, 1, Length@indxNames}], Joined → True,  PlotMarkers → Automatic,
 PlotLegends → indxNames, PlotRange → {{0, 15}, {-0.3, 0.4}}]
```

Out[204]=



In[205]:=
```
Map[#[[2]] &, List2DPlotByIdxNo2[1]]
```

Out[205]=
```
{0., 0.0207009, 0.0425846, 0.0655021, 0.0934042, 0.124873, 0.159299, 0.19851,
 0.241571, 0.290952, 0.351929, 0.426745, 0.517311, 0.630413, 0.774374, 0.966667}
```

In[206]:=
```
res1 = Join[{indxNames}, Transpose@ParallelTable[
     Map[#[[2]] &, List2DPlotByIdxNo2[idxNo]], {idxNo, 1, Length@indxNames}]];
```

In[207]:=
```
res1 // TableForm
```

Out[207]//TableForm=

Geometric Consistency index v2	Geometric Consistency index v1	Koczkodaj's i
0.	0.	0.
0.0207009	0.0651073	0.00160214
0.0425846	0.129267	0.00327629
0.0655021	0.19239	0.00489179
0.0934042	0.25759	0.00692782
0.124873	0.322996	0.00953943
0.159299	0.387617	0.0126891
0.19851	0.452284	0.0160425
0.241571	0.51574	0.0210008
0.290952	0.578656	0.0271097
0.351929	0.643037	0.0365528
0.426745	0.708331	0.0516054
0.517311	0.773241	0.0780539
0.630413	0.838085	0.134161
0.774374	0.902367	0.288618
0.966667	0.966667	0.999933

In[208]:=
```
ParallelTable[Map[#[[2]] &, List2DPlotByIdxNo2[idxNo]],
  {idxNo, 1, Length@indxNames}] // TableForm
```

Out[208]//TableForm=

0.	0.0207009	0.0425846	0.0655021	0.0934042	0.124873	0.
0.	0.0651073	0.129267	0.19239	0.25759	0.322996	0.
0.	0.00160214	0.00327629	0.00489179	0.00692782	0.00953943	0.
0.	0.00190365	0.00374695	0.00589806	0.008786	0.0117399	0.
0.	−0.143392	−0.270586	−0.382372	−0.479947	−0.564795	−0
0.	0.0020522	0.00411613	0.00623764	0.00892141	0.012121	0.
0.	−0.00152817	−0.00356849	−0.00641481	−0.00988332	−0.0143455	−0
0.	0.0214826	0.0434494	0.0684396	0.0967991	0.127007	0.
0.	0.0055594	0.012286	0.0214537	0.0336323	0.0498867	0.
0.	0.0514082	0.104263	0.157962	0.216014	0.275763	0.
0.	0.0189459	0.0394004	0.0601837	0.0868499	0.114556	0.
0.	0.062258	0.123651	0.183621	0.246465	0.308646	0.
0.	0.0610321	0.118195	0.173612	0.231052	0.290873	0.
0.	0.020471	0.0424605	0.0653378	0.093682	0.124904	0.

In[209]:=
```
res2 = Map[Plus @@ Abs /@ # &, ParallelTable[
   Map[#[[2]] &, List2DPlotByIdxNo2[idxNo]], {idxNo, 1, Length@indxNames}]]
```

Out[209]=
```
{4.90483, 7.73337, 1.692, 1.33682, 10.112, 1.58968, 1.7321,
 4.93592, 4.12867, 7.29586, 4.67488, 7.50287, 7.14256, 4.90628}
```

In[210]:=
```
AppendTo[res1, res2] // TableForm
```

Out[210]//TableForm=

Geometric Consistency index v2	Geometric Consistency index v1	Koczkodaj's i
0.	0.	0.
0.0207009	0.0651073	0.00160214
0.0425846	0.129267	0.00327629
0.0655021	0.19239	0.00489179
0.0934042	0.25759	0.00692782
0.124873	0.322996	0.00953943
0.159299	0.387617	0.0126891
0.19851	0.452284	0.0160425
0.241571	0.51574	0.0210008
0.290952	0.578656	0.0271097
0.351929	0.643037	0.0365528
0.426745	0.708331	0.0516054
0.517311	0.773241	0.0780539
0.630413	0.838085	0.134161
0.774374	0.902367	0.288618
0.966667	0.966667	0.999933
4.90483	7.73337	1.692

In[211]:=
```
Map[If[NumberQ@#, NumberForm[#, 4], #] &, Join[res1, {res2}], {2}] // TableForm
```

Out[211]//TableForm=

Geometric Consistency index v2	Geometric Consistency index v1	Koczkodaj's i
0.	0.	0.
0.0207	0.06511	0.001602
0.04258	0.1293	0.003276
0.0655	0.1924	0.004892
0.0934	0.2576	0.006928
0.1249	0.323	0.009539
0.1593	0.3876	0.01269
0.1985	0.4523	0.01604
0.2416	0.5157	0.021
0.291	0.5787	0.02711
0.3519	0.643	0.03655
0.4267	0.7083	0.05161
0.5173	0.7732	0.07805
0.6304	0.8381	0.1342
0.7744	0.9024	0.2886
0.9667	0.9667	0.9999
4.905	7.733	1.692
4.905	7.733	1.692

In[212]:=
```
Export["/tmp/results.xlsx",
 Map[If[NumberQ@#, NumberForm[#, 4], #] &, Join[res1, {res2}], {2}]]
```

Out[212]= /tmp/results.xlsx

In[213]:=
```
SortBy[Transpose[{indxNames, res2}], Last] // TableForm
```

Out[213]//TableForm=

Triad based index 1	1.33682
Triad based alpha index	1.58968
Koczkodaj's index	1.692
Triad based alpha-beta index	1.7321
Salo-Hamalainen index	4.12867
Barzilai's Relative Error index v2	4.67488
Geometric Consistency index v2	4.90483
Oliva et al. index	4.90628
Golden-Wang index	4.93592
iLLS criterion	7.14256
Barzilai's Relative Error index v1	7.29586
Saaty consistency index	7.50287
Geometric Consistency index v1	7.73337
Triad based index 2	10.112