

“By Example” introduction into the Mathematica PairwiseComparisons` Package

Konrad Kułakowski (C) 2014 - 2019
e-mail: konrad@kulakowski.org

The package is available under GNU General Public Licence,
see: <http://www.gnu.org/copyleft/gpl.html>

To install the package on the computer start Mathematica and choose File/Install..., then select PairwiseComparisons.m as the package source and press OK.

To install the package at the Raspberry PI just copy them into /opt/Wolfram/WolframEngine/10.0/AddOns/ExtraPackages and restart the application

Introduction

```
In[139]:= << PairwiseComparisons.wl`;
```

Alternatively you can call:

```
In[140]:= Needs["PairwiseComparisons.wl`"];
```

If you are going to use the package for parallel programming use ParallelNeeds

```
In[141]:= ParallelNeeds["PairwiseComparisons.wl`"];
```

Printing function usage

```
In[142]:= ? GeometricRank
```

```
Out[142]=
```

Symbol

GeometricRank[M] returns rank list given as geometric means of rows of the matrix M



Printing full (with implementation) function usage

In[143]:= ?? GeometricRank

Symbol

GeometricRank[M] returns rank list given as geometric means of rows of the matrix M

Out[143]=

Definitions GeometricRank[PairwiseComparisons`Private`matrix_] :=
(GeometricMean[##1] &)/@PairwiseComparisons`Private`matrix
Full Name PairwiseComparisons`GeometricRank

```
GeometricRank[PairwiseComparisons`Private`matrix_] :=
  (GeometricMean[##1] &)/@PairwiseComparisons`Private`matrix
```

Print all the public functions in the package

In[144]:= ? PairwiseComparisons`*

▼ PairwiseComparisons`

AHPA	HREPartialRank
AHPC	HRERescaledRank
AIJadd	IndexOfDeterminants
AIJgeom	IrreducibleMatrixQ
CardinalToOrdinalPCMatrix	KendallIndex
CircularTriadsList	KendallTauDistance
CircularTriadsNo	KoczkodajConsistentTriad
CIStar	KoczkodajIdx
ConsistentMatrixFromRank	KoczkodajIdxRecip
COP1Check	KoczkodajImproveMatrixStep
COP1ViolationList	KoczkodajTheMostInconsistentTriad
COP2Check	KoczkodajTheWorstTriad
COP2ViolationList	KoczkodajTheWorstTriads
CR	KoczkodajTriadIdx
EigenvalueRank	KoczkodajTriadInconsistency
EigenvalueRankSym	ListOfPossibleRationalEntries
ErrorMatrix	LLSM
EVM	LocalDiscrepancyMatrix
FundamentalScale	MaxInconsistentTriadsNoTies
GCI	MaxInconsistentTriadsWithTies
GED	NGMM
GEDMin	NormalizedKendallTauDistance
GeneralizedGeometricConsistencyIndex1	NumericalRankToOrdinalRank
GeneralizedGeometricConsistencyIndex2	NumericalRankToReverseOrdinalRank
GeneralizedGoldenWangIndex	PrincipalEigenValue
GeneralizedKendallIndex	PrincipalEigenValueSym
GeneralizedKoczkodajIndex	PrincipalEigenVector
GeneralizedRelativeError	RandomBoolean

Out[144]=

GeneralizedRelativeErrorv2	RandomIndex
GeneralizedSaloHamalainenIndex	RandomMatrix
GeneralizedTriadBasedInconsistencyIndexAlpha	RandomMatrixEntry
GeneralizedTriadBasedInconsistencyIndexAlphaBeta	RandomRankingPattern
GeneralizedTupleBasedInconsistencyIndex1	RandomRationalMatrix
GeneralizedTupleBasedInconsistencyIndex2	RandomRationalMatrixEntry
GeometricConsistencyIndex	RandomRationalUniformRankingPattern
GeometricRank	RankOrder
GeometricRescaledRank	RankOrderToOrdinalRank
GGCI1	RankOrderToReverseOrdinalRank
GGCI2	RankOrderWithTies
GGWI	RankToVector
GKI	RationalRandomScaledMatrix
GlobalDiscrepancy	RationalScale
GMM	RC
GoldenWangIndex	RE
GRE	RecreateOrdinalPCMatrix
GREv2	RecreatePCMatrix
GSHI	RelativeError
GTBII1	RelativeErrorv2
GTBII2	RescaleRanking
GTBIIAlpha	REv2
GTBIIAlphaBeta	RI
GWl	SaatyConsistencyRatio
HarkerMatrix	SaatyIdx
HarmonicConsistencyIndex	SaatyIdxSym
HCI	SaloHamalainenIndex
HPCA	SHI
HPCC	SymbPCM
HREConstantTermVector	SymbPCMRec
HREFullRank	TBII1
HREGeomConstantTermVector	TBII2
HREGeomFullRank	TBIIAlpha
HREGeomIntermediateRank	TriadBasedInconsistencyIndex
HREGeomMatrix	TriadBasedInconsistencyIndex2
HREGeomPartialRank	TriadBasedInconsistencyIndexAlpha
HREGeomRescaledRank	VersionPC
HREMatrix	

In[145]=

Assign the matrix to the variable M

$$\text{In[146]:= } M = \begin{pmatrix} 1 & 2 & 3 \\ \frac{1}{2} & 1 & 4 \\ \frac{1}{3} & \frac{1}{4} & 1 \end{pmatrix}$$

$$\text{Out[146]:= } \left\{ \{1, 2, 3\}, \left\{ \frac{1}{2}, 1, 4 \right\}, \left\{ \frac{1}{3}, \frac{1}{4}, 1 \right\} \right\}$$

Eigenvalue based method

To calculate the principal eigenvalue of the matrix M it is enough to call:

`In[147]:= PrincipalEigenValue [M]`

`Out[147]= 3.10785`

Calculate the principal eigenvector of the matrix M

`In[148]:= PrincipalEigenvector [M]`

`Out[148]= {0.806208, 0.558993, 0.193792}`

Calculate the value of the Saaty (eigenvalue based) inconsistency index of M

`In[149]:= SaatyIdx [M]`

`Out[149]= 0.0539237`

Calculate the value of the Saaty consistency ratio of M

`In[150]:= SaatyConsistencyRatio [M]`

`Out[150]= 0.102278`

Calculate Random index for the matrix of size 3 by 3 (note that this value is Monte Carlo approximation)

`In[151]:= RandomIndex [3]`

`Out[151]= 0.527229`

Calculate Random index for the matrix of size 3 by 3 based on 500 random PC matrices with the rational scale 1/9,1/8,...1,2,3,...,9

`In[152]:= RandomIndex [3, 500, 9]`

`Out[152]= 0.521684`

Calculate the eigenvalue based ranking based on M

`In[153]:= EigenvalueRank [M]`

`Out[153]= {0.517134, 0.35856, 0.124306}`

Analytic Hierarchy Process (AHP)

Calculate the rank for the three different criteria using AHP (see Belton and Gear, On a short-coming of Saaty's method of analytic hierarchies, 1983) . To this end we will use two methods AHPA - which implements AHP on the Alternative level, and AHPC which computes the ranking on the criteria

level.

$$\text{In[154]:= } \mathbf{A}_1 = \begin{pmatrix} 1 & 1 & 4 \\ 1 & 1 & 4 \\ 1/4 & 1/4 & 1 \end{pmatrix};$$

In[155]:= EigenvalueRank[A₁]

Out[155]:= {0.444444, 0.444444, 0.111111}

$$\text{In[156]:= } \mathbf{A}_2 = \begin{pmatrix} 1 & 2 & 6 \\ 1/2 & 1 & 3 \\ 1/6 & 1/3 & 1 \end{pmatrix};$$

In[157]:= EigenvalueRank[A₂]

Out[157]:= {0.6, 0.3, 0.1}

$$\text{In[158]:= } \mathbf{A}_3 = \begin{pmatrix} 1 & 1/2 & 1/8 \\ 2 & 1 & 1/4 \\ 8 & 4 & 1 \end{pmatrix};$$

In[159]:= EigenvalueRank[A₃]

Out[159]:= {0.0909091, 0.181818, 0.727273}

$$\text{In[160]:= } \mathbf{A}_{\text{criteria}} = \begin{pmatrix} 1 & 1/2 & 1/4 \\ 2 & 1 & 1/2 \\ 4 & 2 & 1 \end{pmatrix};$$

In[161]:= EigenvalueRank[A_{criteria}]

Out[161]:= {0.142857, 0.285714, 0.571429}

then the result is:

In[162]:= AHPC[A_{criteria}, AHPA[A₁, A₂, A₃]]

Out[162]:= {0.286869, 0.253102, 0.460029}

The above example is taken from the book Matteo Brunelli, Introduction to the Analytic Hierarchy Process, Springer 2015, <http://www.springer.com/gp/book/9783319125015>

Well, of course we may create another level of a hierarchy

$$\text{In[163]:= } \mathbf{A}_{\text{upperLevelCriteria}} = \begin{pmatrix} 1 & 4 & 4 \\ 1/4 & 1 & 3 \\ 1/4 & 1/3 & 1 \end{pmatrix};$$

and compute the AHP ranking for that hierarchy

In[164]:= AHPC[A_{upperLevelCriteria}, AHPC[A_{criteria}, AHPA[A₁, A₂, A₃]]],

AHPC[A_{criteria}, AHPA[A₂, A₃, A₁]], AHPC[A_{criteria}, AHPA[A₃, A₂, A₁]]]

Out[164]:= {0.322511, 0.28831, 0.389179}

Geometric Mean method (Logarithmic Least Square Method)

Calculate the geometric mean based ranking based on M

```
In[165]:= N[GeometricRank[M]]
```

```
Out[165]:= {1.81712, 1.25992, 0.43679}
```

Calculate the geometric mean based ranking based on M rescaled so that the sum of entries is 1

```
In[166]:= N[GeometricRescaledRank[M]]
```

```
Out[166]:= {0.517134, 0.35856, 0.124306}
```

Please note that if you want to have numeric (not symbolic) output the input data also should be “numeric”.

Thus, it is not a bad idea to add a numeric conversion N@ to the input matrix. For example:

```
In[167]:= N[GeometricRank[N@M]]
```

```
Out[167]:= {1.81712, 1.25992, 0.43679}
```

The above applies to all the methods in the package, except eigenvalue based methods such as PrincipalEigenValue,

PrincipalEigenvector, SaatyIdx and EigenvalueRank, which are “by design” numeric.

The geometric consistency index (GCI) as defined in “Aguaron & Moreno-Jimenez, The geometric consistency index: Approximated thresholds“ can be computed by calling:

```
In[168]:= N@GCI[M, GeometricRank[M], 10]
```

```
Out[168]:= 0.0604831
```

Hierarchical Pairwise Comparisons Method (HPCM)

AHPA and AHPC can be generalized to use any ranking method not necessarily EVM. In order to use GMM method instead of EVM one we may use the functions HPCA (Hierarchical Pairwise Comparisons on the level of Alternatives) and HPCC (Hierarchical Pairwise Comparisons on the level of Criteria) respectively.

For example for GMM

```
In[169]:= N@HPCC[GeometricRescaledRank, Acriteria, HPCA[GeometricRescaledRank, A1, A2, A3]]
```

```
Out[169]:= {0.286869, 0.253102, 0.460029}
```

```
In[170]:= N@HPCC[GeometricRescaledRank, AupperLevelCriteria,
```

```
    HPCC[GeometricRescaledRank, Acriteria, HPCA[GeometricRescaledRank, A1, A2, A3]],
```

```
    HPCC[GeometricRescaledRank, Acriteria, HPCA[GeometricRescaledRank, A2, A3, A1]],
```

```
    HPCC[GeometricRescaledRank, Acriteria, HPCA[GeometricRescaledRank, A3, A2, A1]]]
```

```
Out[170]:= {0.322511, 0.28831, 0.389179}
```

We may also mix within the same hierarchy different priority deriving methods. For example:

```
In[171]:= HPCC[GeometricRescaledRank, Acriteria, HPCA[EigenvalueRank, A1, A2, A3]]
```

```
Out[171]:= {0.286869, 0.253102, 0.460029}
```

Heuristic Rating Estimation Method (additive)

Calculate the HRE Matrix for the given matrix M where the unknown concepts are $\{c_1, c_2, c_3\}$ and the known concepts are $\{c_4 = 5, c_5 = 9\}$. It is assumed that the unknown concepts has value 0 whilst the known concepts have the values greater than 0.

Further references could be found in papers :

* Konrad Kułakowski,

Heuristic Rating Estimation Approach to The Pairwise Comparisons Method

<http://arxiv.org/abs/1309.0386>

* Konrad Kułakowski,

A heuristic rating estimation algorithm for the pairwise comparisons method

<http://dx.doi.org/10.1007/s10100-013-0311-x>

SymbPCM[symbol, size] - creates symbolic PC matrix based on the given symbol with the given size

```
In[172]:= M = SymbPCM[m, 5];
```

```
In[173]:= M // MatrixForm
```

```
Out[173]/MatrixForm=
```

$$\begin{pmatrix} 1 & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} \\ m_{2,1} & 1 & m_{2,3} & m_{2,4} & m_{2,5} \\ m_{3,1} & m_{3,2} & 1 & m_{3,4} & m_{3,5} \\ m_{4,1} & m_{4,2} & m_{4,3} & 1 & m_{4,5} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & 1 \end{pmatrix}$$

```
In[174]:= HREMatrix[M,  $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 5 \\ 9 \end{pmatrix}$ ]
```

```
Out[174]=
```

$$\left\{ \left\{ 1, -\frac{m_{1,2}}{4}, -\frac{m_{1,3}}{4} \right\}, \left\{ -\frac{m_{2,1}}{4}, 1, -\frac{m_{2,3}}{4} \right\}, \left\{ -\frac{m_{3,1}}{4}, -\frac{m_{3,2}}{4}, 1 \right\} \right\}$$

Calculate the HRE constant term vector for M

where C_U equals $\{c_1, c_2, c_3\}$ and C_K equals $\{c_4 = 5, c_5 = 9\}$

```
In[175]:= HREConstantTermVector[M,  $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 5 \\ 9 \end{pmatrix}$ ]
```

```
Out[175]=
```

$$\left\{ \left\{ \frac{5 m_{1,4}}{4} + \frac{9 m_{1,5}}{4} \right\}, \left\{ \frac{5 m_{2,4}}{4} + \frac{9 m_{2,5}}{4} \right\}, \left\{ \frac{5 m_{3,4}}{4} + \frac{9 m_{3,5}}{4} \right\} \right\}$$

Auxiliary function that transform an upper triangle matrix into a full and reciprocal matrix

```
In[176]:= M = RecreatePCMatrix [

$$\begin{pmatrix} 1 & \frac{3}{5} & \frac{4}{7} & \frac{5}{8} & \frac{5}{10} \\ 0 & 1 & \frac{5}{7} & \frac{5}{2} & \frac{10}{3} \\ 0 & 0 & 1 & \frac{7}{2} & 4 \\ 0 & 0 & 0 & 1 & \frac{4}{3} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

]
```

```
Out[176]:= {{1,  $\frac{3}{5}$ ,  $\frac{4}{7}$ ,  $\frac{5}{8}$ ,  $\frac{1}{2}$ }, { $\frac{5}{3}$ , 1,  $\frac{5}{7}$ ,  $\frac{5}{2}$ ,  $\frac{10}{3}$ },
{ $\frac{7}{4}$ ,  $\frac{7}{5}$ , 1,  $\frac{7}{2}$ , 4}, { $\frac{8}{5}$ ,  $\frac{2}{5}$ ,  $\frac{2}{7}$ , 1,  $\frac{4}{3}$ }, {2,  $\frac{3}{10}$ ,  $\frac{1}{4}$ ,  $\frac{3}{4}$ , 1}}
```

```
In[177]:=
```

Defining known and unknown alternatives. It is assumed that c_2 and c_3 are known and equal 5 and 7 correspondingly

```
In[178]:= mk =  $\begin{pmatrix} 0 \\ 5 \\ 7 \\ 0 \\ 0 \end{pmatrix}$ 
```

```
Out[178]:= {{0}, {5}, {7}, {0}, {0}}
```

Calculate the HRE ranking vector for unknown alternatives i.e. for c_1 , c_4 and c_5 only

```
In[179]:= mu = N[HREPartialRank[M, mk]]
```

```
Out[179]:= {{2.52765}, {2.88338}, {2.61696}}
```

Calculate the full HRE ranking for the given input matrix M and the vector mk

```
In[180]:= mu = N[HREFullRank[M, mk]]
```

```
Out[180]:= {2.52765, 5., 7., 2.88338, 2.61696}
```

Calculate the full HRE ranking rescaled so that all its entries sum up to 1

```
In[181]:= mu = N[HRERescaledRank[M, mk]]
```

```
Out[181]:= {0.126206, 0.249651, 0.349511, 0.143967, 0.130665}
```

Heuristic Rating Estimation Method (multiplicative/geometric)

Further references could be found in papers:

* Konrad Kułakowski, Grobler-Dębska Katarzyna, Wąs Jarosław, Heuristic rating estimation - geometric approach, <http://arxiv.org/abs/1404.6981>

```
In[182]:= HREGeomMatrix [

$$\begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,5} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & m_{5,5} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ q_4 \\ q_5 \end{pmatrix}$$

]
```

```
Out[182]:= {{4, -1, -1}, {-1, 4, -1}, {-1, -1, 4}}
```


Calculate the HRE constant term vector

```
In[183]:= HREGeomConstantTermVector [ 
$$\begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,5} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & m_{5,5} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ q_4 \\ q_5 \end{pmatrix} ]$$

```

```
Out[183]:= { 
$$\left\{ \frac{\text{Log}[q_4 q_5 m_{1,2} m_{1,3} m_{1,4} m_{1,5}]}{\text{Log}[10]} \right\},$$

```

```
          
$$\left\{ \frac{\text{Log}[q_4 q_5 m_{2,1} m_{2,3} m_{2,4} m_{2,5}]}{\text{Log}[10]} \right\}, \left\{ \frac{\text{Log}[q_4 q_5 m_{3,1} m_{3,2} m_{3,4} m_{3,5}]}{\text{Log}[10]} \right\} }$$

```

Calculate the HRE geometric ranking vector for unknown alternatives i.e. for c_1 , c_4 and c_5 only

```
In[184]:= mu = N[HREGeomPartialRank[M, mk]]
```

```
Out[184]:= {{2.11273}, {2.49035}, {2.13344}}
```

Calculate the full HRE geometric ranking for the given input matrix M and the vector mk

```
In[185]:= mu = N[HREGeomFullRank[M, mk]]
```

```
Out[185]:= {2.11273, 5., 7., 2.49035, 2.13344}
```

Calculate the full HRE geometric ranking, rescaled so that all its entries sum up to 1

```
In[186]:= mu = N[HREGeomRescaledRank[M, mk]]
```

```
Out[186]:= {0.11276, 0.266858, 0.373602, 0.132914, 0.113866}
```

Show intermediate (before raising up to the power) HRE geometric partial rank vector

```
In[187]:= N[HREGeomIntermediateRank[M, mk]]
```

```
Out[187]:= {{0.324843}, {0.396261}, {0.329081}}
```

Bana e Costa and Vansnick's Condition of Order Preservation test

Let calculate the eigenvalue ranking

```
In[188]:= rank = EigenvalueRank[M]
```

```
Out[188]:= {0.119092, 0.27476, 0.356526, 0.130954, 0.118669}
```

.... and check whether the first Bana e Costa and Vansnick condition “condition of order preservation - COP” is satisfied

```
In[189]:= COP1Check[M, mu]
```

```
Out[189]:= True
```

then check whether the second Bana e Costa and Vansnick condition (preserving intensity of preferences postulate) is satisfied

```
In[190]:= COP2Check[M, mu]
```

```
Out[190]:= False
```

Prints the list of pairs for which the 1st COP is not satisfied

```
In[191]:= COP1ViolationList[M, mu]
```

```
Out[191]= {}
```

Prints the list of pairs of pairs for which the 2nd COP is not satisfied

```
In[192]:= COP2ViolationList[M, mu]
```

```
Out[192]= {{False, {{1, 2}, {1, 5}}}, {False, {{1, 2}, {4, 2}}}, {False, {{1, 2}, {5, 2}}},
  {False, {{1, 3}, {1, 5}}}, {False, {{1, 3}, {4, 2}}}, {False, {{1, 3}, {4, 3}}},
  {False, {{1, 3}, {5, 2}}}, {False, {{1, 3}, {5, 3}}}, {False, {{1, 4}, {1, 5}}},
  {False, {{1, 4}, {2, 3}}}, {False, {{1, 5}, {1, 2}}}, {False, {{1, 5}, {1, 3}}},
  {False, {{1, 5}, {1, 4}}}, {False, {{1, 5}, {2, 3}}}, {False, {{1, 5}, {5, 4}}},
  {False, {{2, 1}, {2, 4}}}, {False, {{2, 1}, {2, 5}}}, {False, {{2, 1}, {5, 1}}},
  {False, {{2, 3}, {1, 4}}}, {False, {{2, 3}, {1, 5}}}, {False, {{2, 4}, {2, 1}}},
  {False, {{2, 4}, {3, 1}}}, {False, {{2, 5}, {2, 1}}}, {False, {{2, 5}, {3, 1}}},
  {False, {{3, 1}, {2, 4}}}, {False, {{3, 1}, {2, 5}}}, {False, {{3, 1}, {3, 4}}},
  {False, {{3, 1}, {3, 5}}}, {False, {{3, 1}, {5, 1}}}, {False, {{3, 2}, {4, 1}}},
  {False, {{3, 2}, {5, 1}}}, {False, {{3, 4}, {3, 1}}}, {False, {{3, 5}, {3, 1}}},
  {False, {{4, 1}, {3, 2}}}, {False, {{4, 1}, {5, 1}}}, {False, {{4, 2}, {1, 2}}},
  {False, {{4, 2}, {1, 3}}}, {False, {{4, 3}, {1, 3}}}, {False, {{4, 5}, {5, 1}}},
  {False, {{5, 1}, {2, 1}}}, {False, {{5, 1}, {3, 1}}}, {False, {{5, 1}, {3, 2}}},
  {False, {{5, 1}, {4, 1}}}, {False, {{5, 1}, {4, 5}}}, {False, {{5, 2}, {1, 2}}},
  {False, {{5, 2}, {1, 3}}}, {False, {{5, 3}, {1, 3}}}, {False, {{5, 4}, {1, 5}}}}
```

Koczkodaj's Iterative Inconsistency Reduction algorithm

Calculate the value of the Koczkodaj inconsistency index

```
In[193]:= N[KoczkodajIdx[M]]
```

```
Out[193]= 0.78125
```

Prints the worst Koczkodaj triad in M. As we can see it is $m_{5,3} == \frac{1}{4}$,

$m_{3,1} == \frac{7}{4}$, $m_{5,1} == 2$. The value of inconsistency introduced by this triad is $\frac{25}{32}$

```
In[194]:= KoczkodajTheWorstTriad[M]
```

```
Out[194]= {{5, 3, 1}, {1/4, 7/4, 2}, 25/32}
```

Perform one step of the Koczkodaj inconsistency reduction algorithm. On the output there is a new slightly modified matrix M2 that is expected to be more consistent than M

```
In[195]:= M2 = N[KoczkodajImproveMatrixStep[M]]
```

```
Out[195]= {{1., 0.6, 0.344306, 0.625, 0.829827},
  {1.66667, 1., 0.714286, 2.5, 3.33333}, {2.90439, 1.4, 1., 3.5, 2.41014},
  {1.6, 0.4, 0.285714, 1., 1.33333}, {1.20507, 0.3, 0.414913, 0.75, 1.}}
```

```
In[196]:= KoczkodajIdx[M2]
```

```
Out[196]= 0.585087
```

Group decision making

Let us consider three different PC matrices X , Y and Z that come from three different experts

$$\text{In[197]:= } X = \begin{pmatrix} 1 & x_{12} & x_{13} & x_{14} \\ x_{21} & 1 & x_{23} & x_{24} \\ x_{31} & x_{32} & 1 & x_{34} \\ x_{41} & x_{42} & x_{43} & 1 \end{pmatrix};$$

$$\text{In[198]:= } Y = \begin{pmatrix} 1 & y_{12} & y_{13} & y_{14} \\ y_{21} & 1 & y_{23} & y_{24} \\ y_{31} & y_{32} & 1 & y_{34} \\ y_{41} & y_{42} & y_{43} & 1 \end{pmatrix};$$

$$\text{In[199]:= } Z = \begin{pmatrix} 1 & z_{12} & z_{13} & z_{14} \\ z_{21} & 1 & z_{23} & z_{24} \\ z_{31} & z_{32} & 1 & z_{34} \\ z_{41} & z_{42} & z_{43} & 1 \end{pmatrix};$$

Then it is possible to aggregate the results using appropriate functions.

Aggregate Individual Judgments additively (AIJadd):

`In[200]:= AIJadd[X, Y, Z] // MatrixForm`

`Out[200]/MatrixForm=`

$$\begin{pmatrix} 1 & \frac{1}{3}(x_{12} + y_{12} + z_{12}) & \frac{1}{3}(x_{13} + y_{13} + z_{13}) & \frac{1}{3}(x_{14} + y_{14} + z_{14}) \\ \frac{1}{3}(x_{21} + y_{21} + z_{21}) & 1 & \frac{1}{3}(x_{23} + y_{23} + z_{23}) & \frac{1}{3}(x_{24} + y_{24} + z_{24}) \\ \frac{1}{3}(x_{31} + y_{31} + z_{31}) & \frac{1}{3}(x_{32} + y_{32} + z_{32}) & 1 & \frac{1}{3}(x_{34} + y_{34} + z_{34}) \\ \frac{1}{3}(x_{41} + y_{41} + z_{41}) & \frac{1}{3}(x_{42} + y_{42} + z_{42}) & \frac{1}{3}(x_{43} + y_{43} + z_{43}) & 1 \end{pmatrix}$$

Aggregate Individual Judgments geometrically (AIJgeom)

`In[201]:= AIJgeom[X, Y, Z] // MatrixForm`

`Out[201]/MatrixForm=`

$$\begin{pmatrix} 1 & (x_{12} y_{12} z_{12})^{1/3} & (x_{13} y_{13} z_{13})^{1/3} & (x_{14} y_{14} z_{14})^{1/3} \\ (x_{21} y_{21} z_{21})^{1/3} & 1 & (x_{23} y_{23} z_{23})^{1/3} & (x_{24} y_{24} z_{24})^{1/3} \\ (x_{31} y_{31} z_{31})^{1/3} & (x_{32} y_{32} z_{32})^{1/3} & 1 & (x_{34} y_{34} z_{34})^{1/3} \\ (x_{41} y_{41} z_{41})^{1/3} & (x_{42} y_{42} z_{42})^{1/3} & (x_{43} y_{43} z_{43})^{1/3} & 1 \end{pmatrix}$$

Note that these two functions works also for result lists i.e.:

`In[202]:= AIJgeom[{x1, x2, x3, x4}, {y1, y2, y3, y4}]`

$$\text{Out[202]= } \left\{ \sqrt{x_1 y_1}, \sqrt{x_2 y_2}, \sqrt{x_3 y_3}, \sqrt{x_4 y_4} \right\}$$

`In[203]:= AIJadd[{x1, x2, x3, x4}, {y1, y2, y3, y4}]`

$$\text{Out[203]= } \left\{ \frac{1}{2}(x_1 + y_1), \frac{1}{2}(x_2 + y_2), \frac{1}{2}(x_3 + y_3), \frac{1}{2}(x_4 + y_4) \right\}$$

The aggregated priority vector can also be computed as the result of GED (group euclidian distance) minimization. For example GED for X, Y, Z can be calculated as $\text{GED}[X, Y, Z]$. In this particular case $\text{GED}[X, Y, Z]$ equals:

$$\begin{aligned} \text{In[204]} := & \sqrt{\left(\left(-\frac{w_1}{w_2} + x_{12} \right)^2 + \left(-\frac{w_1}{w_3} + x_{13} \right)^2 + \left(-\frac{w_1}{w_4} + x_{14} \right)^2 + \left(-\frac{w_2}{w_1} + x_{21} \right)^2 + \left(-\frac{w_2}{w_3} + x_{23} \right)^2 + \left(-\frac{w_2}{w_4} + x_{24} \right)^2 + \right. \\ & \left(-\frac{w_3}{w_1} + x_{31} \right)^2 + \left(-\frac{w_3}{w_2} + x_{32} \right)^2 + \left(-\frac{w_3}{w_4} + x_{34} \right)^2 + \left(-\frac{w_4}{w_1} + x_{41} \right)^2 + \left(-\frac{w_4}{w_2} + x_{42} \right)^2 + \left(-\frac{w_4}{w_3} + x_{43} \right)^2 + \\ & \left(-\frac{w_1}{w_2} + y_{12} \right)^2 + \left(-\frac{w_1}{w_3} + y_{13} \right)^2 + \left(-\frac{w_1}{w_4} + y_{14} \right)^2 + \left(-\frac{w_2}{w_1} + y_{21} \right)^2 + \left(-\frac{w_2}{w_3} + y_{23} \right)^2 + \left(-\frac{w_2}{w_4} + y_{24} \right)^2 + \\ & \left(-\frac{w_3}{w_1} + y_{31} \right)^2 + \left(-\frac{w_3}{w_2} + y_{32} \right)^2 + \left(-\frac{w_3}{w_4} + y_{34} \right)^2 + \left(-\frac{w_4}{w_1} + y_{41} \right)^2 + \left(-\frac{w_4}{w_2} + y_{42} \right)^2 + \left(-\frac{w_4}{w_3} + y_{43} \right)^2 + \\ & \left(-\frac{w_1}{w_2} + z_{12} \right)^2 + \left(-\frac{w_1}{w_3} + z_{13} \right)^2 + \left(-\frac{w_1}{w_4} + z_{14} \right)^2 + \left(-\frac{w_2}{w_1} + z_{21} \right)^2 + \left(-\frac{w_2}{w_3} + z_{23} \right)^2 + \left(-\frac{w_2}{w_4} + z_{24} \right)^2 + \\ & \left. \left(-\frac{w_3}{w_1} + z_{31} \right)^2 + \left(-\frac{w_3}{w_2} + z_{32} \right)^2 + \left(-\frac{w_3}{w_4} + z_{34} \right)^2 + \left(-\frac{w_4}{w_1} + z_{41} \right)^2 + \left(-\frac{w_4}{w_2} + z_{42} \right)^2 + \left(-\frac{w_4}{w_3} + z_{43} \right)^2 \right); \end{aligned}$$

Let us consider the following four PC matrices:

$$\text{In[205]} := \mathbf{C1} = \begin{pmatrix} 1 & 2 & \frac{4}{5} & \frac{9}{5} & 9 \\ \frac{1}{2} & 1 & \frac{1}{4} & \frac{7}{4} & 5 \\ \frac{5}{4} & 4 & 1 & \frac{7}{2} & 9 \\ \frac{5}{9} & \frac{4}{7} & \frac{2}{7} & 1 & 5 \\ \frac{1}{9} & \frac{1}{5} & \frac{1}{9} & \frac{1}{5} & 1 \end{pmatrix};$$

$$\text{In[206]} := \mathbf{C2} = \begin{pmatrix} 1 & 2 & \frac{1}{4} & 1 & 9 \\ \frac{1}{2} & 1 & \frac{8}{9} & \frac{2}{3} & 8 \\ 4 & \frac{9}{8} & 1 & 3 & 9 \\ 1 & \frac{3}{2} & \frac{1}{3} & 1 & 9 \\ \frac{1}{9} & \frac{1}{8} & \frac{1}{9} & \frac{1}{9} & 1 \end{pmatrix};$$

$$\text{In[207]} := \mathbf{C3} = \begin{pmatrix} 1 & 1 & \frac{3}{8} & \frac{5}{3} & 9 \\ 1 & 1 & \frac{2}{5} & 1 & 9 \\ \frac{8}{3} & \frac{5}{2} & 1 & 6 & 9 \\ \frac{3}{5} & 1 & \frac{1}{6} & 1 & 5 \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & \frac{1}{5} & 1 \end{pmatrix};$$

$$\text{In[208]} := \mathbf{C4} = \begin{pmatrix} 1 & 1 & \frac{3}{7} & \frac{7}{4} & 5 \\ 1 & 1 & \frac{8}{9} & 1 & 7 \\ \frac{7}{3} & \frac{9}{8} & 1 & 2 & 9 \\ \frac{4}{7} & 1 & \frac{1}{2} & 1 & 5 \\ \frac{1}{5} & \frac{1}{7} & \frac{1}{9} & \frac{1}{5} & 1 \end{pmatrix};$$

The aggregated priority vector minimizing GED can be calculated as follows:

$$\text{In[209]} := \text{GEDMin}[\mathbf{C1}, \mathbf{C2}, \mathbf{C3}, \mathbf{C4}]$$

$$\text{Out[209]} := \{0.254041, 0.224838, 0.313325, 0.175409, 0.0323872\}$$

For minimization a differential evolution procedure is used.

Let us look, for example, at the same matrices aggregated using AIJ + GMM

```
In[210]:= GeometricRescaledRank@N@AIJgeom[C1, C2, C3, C4]
Out[210]:= {0.224222, 0.18592, 0.404349, 0.154636, 0.0308722}
```

Incomplete Pairwise Comparisons Matrix - Harker Method

The idea comes from P. T. Harker, Alternative Modes of Questioning in The Analytic Hierarchy Process, Math Modeling, 1987. The

Let M be an incomplete pairwise comparisons matrix.

```
In[211]:= M =  $\begin{pmatrix} 1 & 2 & \square \\ 1/2 & 1 & 2 \\ \square & 1/2 & 1 \end{pmatrix};$ 
```

Thus, to compute the ranking based on M we need to compute matrix A (hereinafter referred to as Harker matrix)

```
In[212]:= A = HarkerMatrix[M];
```

```
In[213]:= A // MatrixForm
```

```
Out[213]//MatrixForm=
 $\begin{pmatrix} 2 & 2 & 0 \\ \frac{1}{2} & 1 & 2 \\ 0 & \frac{1}{2} & 2 \end{pmatrix}$ 
```

Then compute the eigenvalue based ranking and inconsistency index as usual:

```
In[214]:= EigenvalueRank[A]
```

```
Out[214]:= {0.571429, 0.285714, 0.142857}
```

```
In[215]:= SaatyIdx[A]
```

```
Out[215]:= 0
```

The ranking can be computed for incomplete matrix only if it is irreducible, i.e. associated adjacency graph is irreducible. To check whether matrix is irreducible we can use IrreducibleMatrixQ

```
In[216]:= IrreducibleMatrixQ[M]
```

```
Out[216]:= True
```

As it easy to observe, the Harker method works for both complete and incomplete matrices. For complete matrices it is just a EVM method. For this reason there is another, general function $EVM[M]$ that can handle both complete and incomplete matrices.

```
In[217]:= EVM[M]
```

```
Out[217]:= {0.571429, 0.285714, 0.142857}
```

Incomplete Pairwise Comparisons Matrix - Geometric Mean Method for Incomplete matrices

The idea comes from Sandor et. al, On optimal completion of incomplete pairwise comparison matrices, Mathematical and Computer Modelling, 2010. Note that this method works for both complete and incomplete matrices. When the matrix is complete it simply boils down to GMM method.

Let M be an incomplete pairwise comparisons matrix.

$$\text{In[218]:= } \mathbf{M} = \begin{pmatrix} 1 & 2 & 3 & \square & \square \\ 1/2 & 1 & 2 & \square & \square \\ 1/3 & 1/2 & 1 & 2 & \square \\ \square & \square & 1/2 & 1 & 3 \\ \square & \square & \square & 1/3 & 1 \end{pmatrix};$$

Thus, to compute the GMM ranking based on M we need to call GMM[M]

In[219]:= GMM [M]

$$\begin{aligned} \text{Out[219]= } & \left\{ e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 44 \text{Log}[6])} \right. / \\ & \left(e^{\frac{1}{25} (8 \text{Log}[\frac{3}{2}] - 21 \text{Log}[3] - 12 \text{Log}[6])} + e^{\frac{1}{25} (13 \text{Log}[\frac{3}{2}] - 6 \text{Log}[3] - 7 \text{Log}[6])} + e^{\frac{1}{25} (-2 \text{Log}[\frac{3}{2}] - \text{Log}[3] + 3 \text{Log}[6])} + \right. \\ & \left. e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 19 \text{Log}[6])} + e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 44 \text{Log}[6])} \right), \\ & e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 19 \text{Log}[6])} / \left(e^{\frac{1}{25} (8 \text{Log}[\frac{3}{2}] - 21 \text{Log}[3] - 12 \text{Log}[6])} + e^{\frac{1}{25} (13 \text{Log}[\frac{3}{2}] - 6 \text{Log}[3] - 7 \text{Log}[6])} + \right. \\ & \left. e^{\frac{1}{25} (-2 \text{Log}[\frac{3}{2}] - \text{Log}[3] + 3 \text{Log}[6])} + e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 19 \text{Log}[6])} + \right. \\ & \left. e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 44 \text{Log}[6])} \right), e^{\frac{1}{25} (-2 \text{Log}[\frac{3}{2}] - \text{Log}[3] + 3 \text{Log}[6])} / \\ & \left(e^{\frac{1}{25} (8 \text{Log}[\frac{3}{2}] - 21 \text{Log}[3] - 12 \text{Log}[6])} + e^{\frac{1}{25} (13 \text{Log}[\frac{3}{2}] - 6 \text{Log}[3] - 7 \text{Log}[6])} + e^{\frac{1}{25} (-2 \text{Log}[\frac{3}{2}] - \text{Log}[3] + 3 \text{Log}[6])} + \right. \\ & \left. e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 19 \text{Log}[6])} + e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 44 \text{Log}[6])} \right), \\ & e^{\frac{1}{25} (13 \text{Log}[\frac{3}{2}] - 6 \text{Log}[3] - 7 \text{Log}[6])} / \left(e^{\frac{1}{25} (8 \text{Log}[\frac{3}{2}] - 21 \text{Log}[3] - 12 \text{Log}[6])} + e^{\frac{1}{25} (13 \text{Log}[\frac{3}{2}] - 6 \text{Log}[3] - 7 \text{Log}[6])} + \right. \\ & \left. e^{\frac{1}{25} (-2 \text{Log}[\frac{3}{2}] - \text{Log}[3] + 3 \text{Log}[6])} + e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 19 \text{Log}[6])} + \right. \\ & \left. e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 44 \text{Log}[6])} \right), e^{\frac{1}{25} (8 \text{Log}[\frac{3}{2}] - 21 \text{Log}[3] - 12 \text{Log}[6])} / \\ & \left(e^{\frac{1}{25} (8 \text{Log}[\frac{3}{2}] - 21 \text{Log}[3] - 12 \text{Log}[6])} + e^{\frac{1}{25} (13 \text{Log}[\frac{3}{2}] - 6 \text{Log}[3] - 7 \text{Log}[6])} + e^{\frac{1}{25} (-2 \text{Log}[\frac{3}{2}] - \text{Log}[3] + 3 \text{Log}[6])} + \right. \\ & \left. e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 19 \text{Log}[6])} + e^{\frac{1}{75} (-21 \text{Log}[\frac{3}{2}] + 27 \text{Log}[3] + 44 \text{Log}[6])} \right) \left. \right\} \end{aligned}$$

There is also a numerical shorthand for this function:

In[220]:= NGMM [M]

Out[220]= {0.4866, 0.267786, 0.147368, 0.0736842, 0.0245614}

This functions also works for complete matrices, and by definition, they are equal to GMM method

$$\text{In[221]:= } \mathbf{M} = \begin{pmatrix} 1 & 2 & 3 & 1 & 2 \\ 1/2 & 1 & 2 & 1 & 1 \\ 1/3 & 1/2 & 1 & 2 & 1 \\ 1 & 1 & 1/2 & 1 & 3 \\ 1/2 & 1 & 1 & 1/3 & 1 \end{pmatrix};$$

```
In[222]:= N@GMM[M] == GeometricRescaledRank[M]
```

```
Out[222]:= True
```

The ranking errors and the ranking discrepancies

Let the $w = \{w_1, w_2, w_3\}$ be the ranking vector whilst M a PC matrix. An error is defined as $e_{ij} = m_{ij}(w_j/w_i)$ and it corresponds to the “discrepancy” between individual judgment m_{ij} and the ranking result. Hence, the error matrix is just $E = [e_{ij}]$. E.g.:

```
In[223]:= M = RecreatePCMatrix [
  (
    (
      1 3/5 4/7 5/8 5/10
      0 1 5/7 5/2 10/3
      0 0 1 7/2 4
      0 0 0 1 4/3
      0 0 0 0 1
    )
  )
];
```

```
In[224]:= ErrorMatrix[M, EigenvalueRank[M]] // MatrixForm
```

```
Out[224]/MatrixForm=
```

$$\begin{pmatrix} 1. & 0.722398 & 0.584559 & 1.45507 & 2.00712 \\ 1.38428 & 1. & 1.07892 & 0.839258 & 0.694602 \\ 1.71069 & 0.92685 & 1. & 0.777866 & 0.751091 \\ 0.687253 & 1.19153 & 1.28557 & 1. & 0.827639 \\ 0.498227 & 1.43967 & 1.3314 & 1.20826 & 1. \end{pmatrix}$$

The error matrix is not symmetric. An attempt to symmetrization of error matrix leads to the local discrepancy matrix $D = (d_{ij})$ where $d_{ij} = \max\{e_{ij} - 1, 1/e_{ij} - 1\}$. Another interesting property of the local discrepancy matrix is the fact that wherever $d_{ij} = 0$ this means that the local discrepancy (error) is 0. If $d_{ij} = X$ this means that the local judgment m_{ij} differs from the ratio $\frac{w_i}{w_j}$ by $100\% \times X$

```
In[225]:= Chop@LocalDiscrepancyMatrix[M, EigenvalueRank[M]] // MatrixForm
```

```
Out[225]/MatrixForm=
```

$$\begin{pmatrix} 0 & 0.384278 & 0.71069 & 0.455068 & 1.00712 \\ 0.384278 & 0 & 0.0789237 & 0.191529 & 0.439673 \\ 0.71069 & 0.0789237 & 0 & 0.285569 & 0.331397 \\ 0.455068 & 0.191529 & 0.285569 & 0 & 0.208257 \\ 1.00712 & 0.439673 & 0.331397 & 0.208257 & 0 \end{pmatrix}$$

The greatest entry of the local discrepancy matrix can be found by using the GlobalDiscrepancy function

```
In[226]:= GlobalDiscrepancy[M, EigenvalueRank[M]]
```

```
Out[226]:= 1.00712
```

Usefull methods built into Mathematica

Spearman Rank Correlation (compare with the example from https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient)

```
In[227]:= Needs["MultivariateStatistics`"];
```

```
In[228]:= N@SpearmanRankCorrelation[{86, 97, 99, 100, 101, 103, 106, 110, 112, 113},
  {0, 20, 28, 27, 50, 29, 7, 17, 6, 12}]
```

```
Out[228]:= -0.175758
```

```
In[229]:=  $r_s = \text{N@SpearmanRankCorrelation}[\{1.1, 1.57, 0.51, 1.1, 1.1\}, \{1.2, 1, 2.3, 1, 18\}]$ 
```

```
Out[229]:= -0.573539
```

```
In[230]:= n = Length@{1.1, 1.57, 0.51, 1.1, 1.1};
```

```
In[231]:=  $t = N[r_s * \text{Sqrt}[\frac{n-2}{1-r_s^2}]]$ 
```

```
Out[231]:= -1.21268
```

Tau Corellation

```
In[232]:= KendallTau[{86, 97, 99, 100, 101, 103, 106, 110, 112, 113},
  {0, 20, 28, 27, 50, 29, 7, 17, 6, 12}]
```

```
Out[232]:= - $\frac{1}{9}$ 
```

Rank Order (with and without Ties)

The use of RankOrder[ranking] is a way to shift from cardinal ranking as produced by EVM or GMM to the ordinal ranking. Hence, the result is a list of alternatives ordered according to the outrank relation $<$ introduced by the cardinal ranking. I.e. for two alternatives c_i and c_j it will holds that $c_i < c_j$ if $w(c_i) < w(c_j)$ and, of course $c_i \sim c_j$ if $w(c_i) = w(c_j)$. For example:

```
In[233]:= ranking = EigenvalueRank[M]
```

```
Out[233]:= {0.119092, 0.27476, 0.356526, 0.130954, 0.118669}
```

```
In[234]:= RankOrder[ranking]
```

```
Out[234]:= {3, 2, 4, 1, 5}
```

The above result means that the highest priority has 3rd alternative, then goes 2nd alternative, 4th, 1st and 5th at the end. It may happen, however, that some alternatives have the same priority. E.g.

```
In[235]:= ranking = EigenvalueRank[ $\begin{pmatrix} 1 & 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1/3 & 1 & 1 & 1 & 1 \end{pmatrix}$ ]
```

```
Out[235]:= {0.256633, 0.194146, 0.194146, 0.194146, 0.16093}
```

In such a case RankOrderWithTies[] groups the alternatives with the similar priorities

```
In[236]:= RankOrderWithTies[ranking]
```

```
Out[236]:= {1, 2, 4, 3, 5}
```

The above result means that the alternatives 2, 3, and 4 took ex aequo the second position.

Sometimes it is useful to use ordinal ranking instead of cardinal one. In the ordinal ranking vector every alternative gets weight which is its position in the numerical ranking. To compute ordinal ranking from rank order we may use `RankOrderToOrdinalRank`

```
In[237]:= RankOrderToOrdinalRank[{1, {2, 3, 4}, 5}]
```

```
Out[237]:= {1, 2, 2, 2, 3}
```

The same result we may get directly from the numerical ranking by using `NumericalRankToOrdinalRank`

```
In[238]:= NumericalRankToOrdinalRank[ranking]
```

```
Out[238]:= {1, 2, 4, 3, 5}
```

The only inconvenience of the above method lies in the fact that the most important first alternative gets the lowest weight (first place). To remove this inconvenience we may want to assign to the alternatives their positions in the ranking in the reverse order. For this purpose we can use `RankOrderToReverseOrdinalRank`.

```
In[239]:= RankOrderToReverseOrdinalRank[{1, {2, 3, 4}, 5}]
```

```
Out[239]:= {3, 2, 2, 2, 1}
```

and similarly

```
In[240]:= NumericalRankToReverseOrdinalRank[ranking]
```

```
Out[240]:= {5, 4, 2, 3, 1}
```

Montecarlo PC Matrices

Sometimes it is useful to generate random PC matrix to verify some properties of such matrix statistically

For example let us create some real random ranking of 7 alternatives with values from interval $[1/15, 15]$:

```
In[241]:= ranking = RandomRankingPattern[7, 15]
```

```
Out[241]:= {0.67423, 13.7385, 0.468824, 0.209944, 0.645864, 0.996717, 0.145665}
```

Then we may create fully consistent matrix based on this ranking by setting $c_{ij} = w_i/w_j$

```
In[242]:= M = RandomMatrix[7, ranking, 1];
```

```
In[243]:= M // MatrixForm
```

```
Out[243]/MatrixForm=
```

$$\begin{pmatrix} 1. & 0.049076 & 1.43813 & 3.21147 & 1.04392 & 0.676451 & 4.62864 \\ 20.3766 & 1. & 29.3041 & 65.4387 & 21.2715 & 13.7837 & 94.3157 \\ 0.695348 & 0.0341249 & 1. & 2.23309 & 0.725888 & 0.470369 & 3.21851 \\ 0.311384 & 0.0152815 & 0.44781 & 1. & 0.32506 & 0.210636 & 1.44128 \\ 0.957927 & 0.0470113 & 1.37762 & 3.07636 & 1. & 0.647991 & 4.4339 \\ 1.4783 & 0.0725492 & 2.12599 & 4.74753 & 1.54323 & 1. & 6.84253 \\ 0.216046 & 0.0106027 & 0.310703 & 0.693826 & 0.225535 & 0.146145 & 1. \end{pmatrix}$$

```
In[244]:= SaatyIdx[M]
```

```
Out[244]= 0
```

We may also disturb every entry by a factor chosen from the range $[1/1.3, 1.3]$. This makes the random matrix a bit inconsistent:

```
In[245]:= M = RandomMatrix[7, ranking, 1.3];
```

```
In[246]:= M // MatrixForm
```

```
Out[246]/MatrixForm=
```

$$\begin{pmatrix} 1. & 0.0445912 & 1.50957 & 3.41255 & 0.95933 & 0.616164 & 3.91523 \\ 22.426 & 1. & 33.7968 & 59.7251 & 22.592 & 17.2469 & 91.6729 \\ 0.662442 & 0.0295886 & 1. & 1.75753 & 0.865142 & 0.605392 & 3.3743 \\ 0.293036 & 0.0167434 & 0.568981 & 1. & 0.262261 & 0.230329 & 1.29289 \\ 1.04239 & 0.0442634 & 1.15588 & 3.81299 & 1. & 0.807267 & 4.15915 \\ 1.62294 & 0.0579815 & 1.65182 & 4.34161 & 1.23875 & 1. & 5.901 \\ 0.255413 & 0.0109083 & 0.296358 & 0.773461 & 0.240433 & 0.169463 & 1. \end{pmatrix}$$

```
In[247]:= SaatyIdx[M]
```

```
Out[247]= 0.00617334
```

```
In[248]:= KoczkodajIdx[M]
```

```
Out[248]= 0.467218
```

One may want to consider disturbed rational matrix. To this end first we create rational ranking pattern of 7 alternatives uniformly distributed over the scale 1/15 to 15:

```
In[249]:= ranking = RandomRationalUniformRankingPattern[7, 15]
```

```
Out[249]= {9, 10,  $\frac{1}{9}$ , 14, 12, 7,  $\frac{1}{15}$ }
```

Then we create fully consistent Rational PC matrix over this ranking

```
In[250]:= M = RandomRationalMatrix[7, 15 * 15, ranking, 1];
```

```
In[251]:= M // MatrixForm
```

```
Out[251]/MatrixForm=
```

$$\begin{pmatrix} 1 & \frac{9}{10} & 81 & \frac{9}{14} & \frac{3}{4} & \frac{9}{7} & 135 \\ \frac{10}{9} & 1 & 90 & \frac{5}{7} & \frac{5}{6} & \frac{10}{7} & 150 \\ \frac{1}{81} & \frac{1}{90} & 1 & \frac{1}{126} & \frac{1}{108} & \frac{1}{63} & \frac{5}{3} \\ \frac{14}{9} & \frac{7}{5} & 126 & 1 & \frac{7}{6} & 2 & 210 \\ \frac{4}{3} & \frac{6}{5} & 108 & \frac{6}{7} & 1 & \frac{12}{7} & 180 \\ \frac{7}{9} & \frac{7}{10} & 63 & \frac{1}{2} & \frac{7}{12} & 1 & 105 \\ \frac{1}{135} & \frac{1}{150} & \frac{3}{5} & \frac{1}{210} & \frac{1}{180} & \frac{1}{105} & 1 \end{pmatrix}$$

```
In[252]:= SaatyIdx[M]
```

```
Out[252]= 0
```

```
In[253]:= KoczkodajIdx[M]
```

```
Out[253]= 0
```

As previously we may also disturb every entry by the factor chosen from the range $[1/1.3, 1.3]$, however, this time we have also to find the closest rational number within the set of numbers in the form $\frac{p}{q}$ where p, q belongs to $\{1, 2, 3, \dots, 15^2\}$. To get this matrix we have to call once again:

```
In[254]:= M = RandomRationalMatrix[7, 15 * 15, ranking, 1.3];
```

```
In[255]:= M // MatrixForm
```

```
Out[255]/MatrixForm=
```

$$\begin{pmatrix} 1 & \frac{31}{41} & 95 & \frac{85}{159} & \frac{172}{197} & \frac{137}{97} & 134 \\ \frac{41}{31} & 1 & 84 & \frac{22}{31} & \frac{151}{192} & \frac{41}{35} & 167 \\ \frac{1}{95} & \frac{1}{84} & 1 & \frac{1}{143} & \frac{1}{107} & \frac{3}{208} & \frac{181}{106} \\ \frac{159}{85} & \frac{31}{22} & 143 & 1 & \frac{57}{41} & \frac{198}{85} & 206 \\ \frac{197}{172} & \frac{192}{151} & 107 & \frac{41}{57} & 1 & \frac{205}{94} & 147 \\ \frac{97}{137} & \frac{35}{41} & \frac{208}{3} & \frac{85}{198} & \frac{94}{205} & 1 & 132 \\ \frac{1}{134} & \frac{1}{167} & \frac{106}{181} & \frac{1}{206} & \frac{1}{147} & \frac{1}{132} & 1 \end{pmatrix}$$

but of course this time there is some small inconsistency

```
In[256]:= SaatyIdx[M]
```

```
Out[256]= 0.00798485
```

```
In[257]:= N@KoczkodajIdx[M]
```

```
Out[257]= 0.489357
```

Both methods for generating random PC matrices has its short forms for default scales and patterns. They need only two parameters: size of the matrix and disturbance level

```
In[258]:= M = RandomMatrix[5, 1.5];
```

```
In[259]:= M // MatrixForm
```

```
Out[259]/MatrixForm=
```

$$\begin{pmatrix} 1. & 0.581627 & 0.0627008 & 0.0982947 & 0.259913 \\ 1.71931 & 1. & 0.0762267 & 0.14598 & 0.355001 \\ 15.9488 & 13.1188 & 1. & 1.62351 & 2.09754 \\ 10.1735 & 6.85027 & 0.61595 & 1. & 1.19175 \\ 3.84744 & 2.8169 & 0.476749 & 0.839102 & 1. \end{pmatrix}$$

```
In[260]:= M = RandomRationalMatrix[5, 1.5];
```

```
In[261]:= M // MatrixForm
```

```
Out[261]/MatrixForm=
```

$$\begin{pmatrix} 1 & \frac{7}{4} & \frac{1}{7} & 9 & 9 \\ \frac{4}{7} & 1 & \frac{1}{9} & 7 & \frac{9}{2} \\ 7 & 9 & 1 & 9 & 9 \\ \frac{1}{9} & \frac{1}{7} & \frac{1}{9} & 1 & 1 \\ \frac{1}{9} & \frac{2}{9} & \frac{1}{9} & 1 & 1 \end{pmatrix}$$

For example is we want to check what is the average inconsistency of the rational matrices 5 by 5 over the fundamental scale disturbed by the factor from $[1/2, 2]$ we may just call:

```
In[262]:= Mean@(SaatyIdx /@ Table[RandomRationalMatrix[5, 2], {100}])
```

```
Out[262]= 0.0731469
```

The above result was computed based on 100 randomly chosen rational random PC matrices.

The fully random rational matrix based on the rational scale can be obtained by RationalRandomS-

caledMatrix[size of a matrix, size of a scale]:

```
In[263]:= M = RationalRandomScaledMatrix[7, 9];
```

```
In[264]:= M // MatrixForm
```

Out[264]/MatrixForm=

$$\begin{pmatrix} 1 & \frac{1}{6} & \frac{1}{9} & 5 & 8 & \frac{1}{4} & 7 \\ 6 & 1 & 8 & 3 & 5 & 2 & \frac{1}{9} \\ 9 & \frac{1}{8} & 1 & \frac{1}{9} & \frac{1}{9} & \frac{1}{7} & 4 \\ \frac{1}{5} & \frac{1}{3} & 9 & 1 & 7 & 1 & \frac{1}{3} \\ \frac{1}{8} & \frac{1}{5} & 9 & \frac{1}{7} & 1 & \frac{1}{3} & \frac{1}{3} \\ 4 & \frac{1}{2} & 7 & 1 & 3 & 1 & \frac{1}{9} \\ \frac{1}{7} & 9 & \frac{1}{4} & 3 & 3 & 9 & 1 \end{pmatrix}$$