Orthosim Target-Comparison Matrix Fitting v3.0

Paul Barrett
Advanced Projects R&D Ltd.
Auckland
New Zealand

email: paul@pbarrett.net
Web: www.pbarrett.net

30th December, 2019

Table of Contents

Part

I

1 Introduction

1.1 Comparing matrix/coordinate data to a Target Matrix

When an investigator wishes to compare a matrix of factor loadings or multidimensional scaling (MDS) coordinates to those from another such analysis, given the same variables have been used in both analyses and where the coordinates/factors are orthogonal to one another, then this program will be of utility. The program calculates four different kinds of similarity coefficient between a comparison and target matrix - both "as they are" on input to the program, and after the comparison matrix has been orthogonally rotated against the target matrix so as to minimize the sum of squared deviations between the comparison matrix and target matrix values. For exploratory factor analysis (EFA) vectors (factors), the program allows an investigator to choose between two kinds of approaches to the problem - that using a "procrustes" approach whereby the matrices are expressed in the same unit-metric space (coordinates of -1 and +1 , irrespective of the initial sizes of the loadings/coordinate values - they are stretched or shrunk accordingly so as to occupy a joint unified metric space), and a non-procrustes approach whereby the comparison and target matrices remain unadjusted, and the comparison matrix is rotated against the target "as is". Both orthogonal or obliquely rotated matrices may be presented to the program. However, oblique matrices will be "transformed" to orthogonal ones, prior to the factor comparison methodology being applied to these orthogonal versions.

When submitting Multidimensional Scaling vectors (coordinate dimensions) for comparison, both matrices are initially centered (their coordinate-space origins are equated), are row-normalized (the "procrustes" approach which expresses each matrix in a normalized unit metric space which preserves the distance relations), then any coordinate "reflections" undone as part of the orthogonal rotation to maximum congruity.This is known as "**configural similarity**" (Borg and Groenen, 1997). The reason for these specific transformations is that MDS solutions are arbitrary in terms of their location, scale, and orientation of variables in geometric space.It is the distance relations between variables which are critical in MDS; such relations can be preserved while allowing the origin, scale, and reflection of solutions to vary. Hence, the extra transformations required prior to congruential rotation.

The EFA-specific procedures of the program are of use for exploratory factor comparison analyses, or where the number of variables is so large as to preclude a structural equation modeling/confirmatory factor analysis approach. or where an investigator wishes to simply rotate a set of data to a schematic target (1s and 0s as loadings/coordinate values). The **Validimax** routine published in McCrae and Costa (1989.,1994) and used in McCrae, Zonderman, Bond, Costa, & Paunonen (1996) is the same as the **Orthosim** non-procrustes routine, which is the **Kaiser, Hunka, Bianchini** (1971) algorithm without the initial row-normalisation of the matrices. All are based upon Schonemann's (1966) and Cliff's (1966) expositions, which in turn are based upon even earlier work such as Ahmavarra (1954).

The key feature to remember is that this routine utilizes least squares fit of the entire comparison matrix to a target matrix. It does not do so one factor at a time - but rather rotates the entire orthogonal comparison solution in m-dimensional space (where m = the number of factors or MDS coordinate dimensions.) toward the target solution. Evans (1971) provides a nice overview of the methodologies proposed for comparison to target matrix model fitting.

References

Ahmavaara, Y. (1954) Transformation analysis of factorial data. *Annales Academiae Scientarum Fennical*, Series B, 88, 2, 1-164.

Borg, I., and Groenen, P. (1997) *Modern Multidimensional Scaling: Theory and Applications*. Springer. ISBN: 0-387-94845-7.

Cliff. N. (1966) Orthogonal rotation to congruence. *Psychometrika*, 31, 1, 33-42.

Evans, G.T. (1971) Transformation of factor matrices to achieve congruence. *British Journal of Mathematical and Statistical Psychology*, 24, 22-48.

Kaiser, H. F., Hunka, S., Bianchini, J. C. (1971) Relating factors between studies based upon different individuals. *Multivariate Behavioral Research*, 409-422.

McCrae, Robert R; Costa, Paul T. Rotation to maximize the construct validity of factors in the NEO Personality Inventory. *Multivariate Behavioral Research. Vol 24(1) Jan 1989, 107-124.*

McCrae, Robert R; Costa, Paul T. "Rotation to maximize the construct validity of factors in the NEO Personality Inventory": Erratum. *Multivariate Behavioral Research. Vol 29(2) 1994, [126].*
Reports an error in the original article by R. R. McCrae and P. T. Costa (Multivariate Behavioral Research, 1989[Jan], Vol 24[1], 107-224). On page 115, in Table 1, the loading of Conscientiousness of the Varimax Neuroticism factor should be -.31.

McCrae, R. R., Zonderman, A. B., Bond, M. H., Costa, P. T. Jr., Paunonen, S. V. (1996) Evaluating replicability of factors in the revised NEO Personality Inventory: Confirmatory factor analysis versus procrustes rotation. *Journal of Personality and Social Psychology*, 70, 3, 552-566.

Schonemann, P. (1966) A general solution of the orthogonal procrustes problem. Psychometrika, 31, 1-10.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

1.2 The Procrustes Method

In Greek mythology, Procrustes was an ancient innkeeper/bandit who stood by the roadside, beckoning all travelers to come in for a good meal and a night's rest on his legendary bed—a bed which could accommodate anyone, short and tall alike. Once you had decided to take him up on his offer and were well fed, he conducted you to his vaunted bedchamber, whereupon you would be placed on his one-size-fits-all iron bed. If you were too long for his bed, he cut your legs off. If you were too short, he stretched you out on the rack.

From Wikipedia ...
In Greek mythology, Procrustes ("the stretcher") was a bandit from Attica killed by Theseus. In Eleusis, he had a bed which he invited passersby to lie down in. When they did so, he either stretched them or cut off body parts to make them fit into the bed. He continued this until he encountered Theseus and ended up in his own bed.Killing Procrustes is the last adventure of Theseus. Theseus pushed Procrustes on his own bed and chopped off his head and feet.Procrustes was a nickname for Damastus, Procoptas, or Polypemon.

Any attempt to reduce men to one standard, one way of thinking, or one way of acting, is called placing them on Procrustes' bed, and the person who makes the attempt is called Procrustes.
Ovid VII, 438. The moral of the story is used as a metaphor for life: I make my own bed, and lie in it.

Procrustes analysis is the name for the process of performing a shape preserving Euclidean transformation to a set of shapes. This removes variations in translation, rotation and scaling across the dataset, so to move them into a common frame of reference. This is generally the precursor to further statistical analysis.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

1.3 The Orthosim-v3 program

Orthosim Features

1. Orthogonal Congruential rotation for EFA factor vectors - with or without row normalisation (i.e. no Procrustes effect). The Orthosim *Procrustes* solution is the original Kaiser, Hunka, Bianichini algorithm which uses row normalization prior to configural matching (albeit modified by Barrett et al (1998) in the light of criticisms by Ten Berge (1996)).
2. Configural Similarity assessment for MDS coordinate dimension vectors.

3. Input matrices printed as standard.
4. Two levels of output - brief and long.
5. Computes the Root-Mean-Square Difference between input comparison and target matrices
6. Computes the Root-Mean-Square Difference between optimally rotated comparison and target matrices
7. Computes the overall solution congruence
8. Computes the overall solution double-scaled euclidean distance similarity (DSED) coefficient
9. Computes the overall solution kernel smoothed distance (KSD) similarity coefficient
10. Computes variable congruence similarity indices
11. Computes Factor/Coordinate DSED and KSD comparisons between the optimally rotated comparison and target matrices.
12. Computes Factor/Coordinate comparison matrices using Pearson and Congruence coefficients between the optimally rotated comparison and target matrices.
13. Computes Factor/Coordinate comparison matrices using Pearson and Congruence coefficients between the initial comparison and target matrices.

Important

- The limits of the program are 350 variables and 35 factors/MDS coordinate dimensions per matrix
- The program can cope with unequal numbers of factors – but, the target matrix must contain the higher number of factors.
- **Both matrices must have the same number of variables**.

Especially Important!

Unless the target matrix was a Varimax/orthogonal rotated matrix – an obliquely rotated input comparison and/or target matrix gets "orthogonalised" before factor/MDS coordinate matching – which means mostly it is uninterpretable (as it is just an arbitrary orthogonalised version of the obliquely rotated matrix. This is because the obliquely rotated matrices are "orthogonalised" to an arbitrary orthogonal representation (not simple structure at all) – then the orthogonalised comparison matrix is compared to the orthogonalised target matrix. Remember, the point of the KHB/orthogonal congruential rotation is to enable you to say how similar two orthogonal sets of vectors are to one another, **prior to any rotational solution**. In reality, the maximally congruent matrix output is only of value when using an orthogonal target matrix as input, and not using row-normalisation (procrustean stretching/shrinking). This is because the already orthogonal matrix remains unchanged (untransformed) as the target matrix– and the comparison matrix is then rigidly rotated toward it. The comparison matrix could of course be obliquely rotated – indeed, it could be the same orthogonal target matrix, obliquely rotated, so that you can see how similar the obliquely rotated factor loadings are to the orthogonal version factor loadings, as well as confirming the congruential congruence via the KHB coefficients.

References

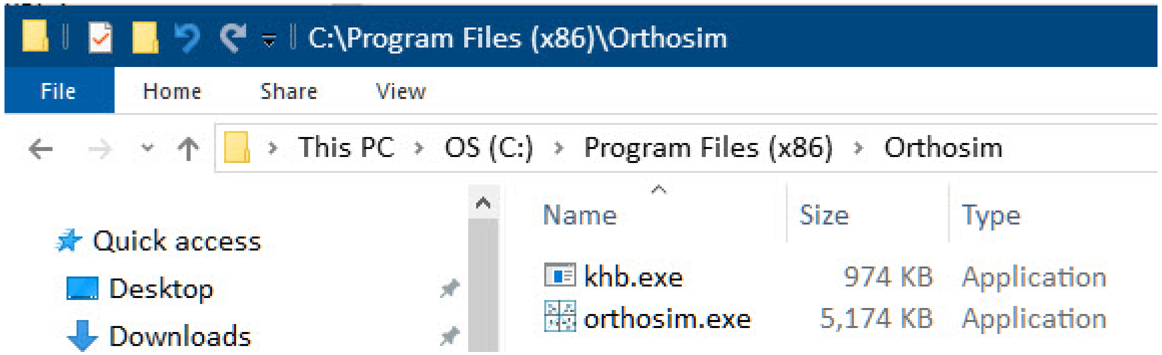
Barrett, P. T., Petrides, K. V., Eysenck, S. B. G., & Eysenck, H. J. (1998) The Eysenck Personality Questionnaire: An examination of the factorial similarity of P, E, N, and L across 34 countries. *Personality and Individual Differences*, 25, 5, 805-819.

Ten Berge, J. M. F. (1996) The Kaiser, Hunka and Bianchini factor similarity coefficients: A cautionary note. *Multivariate Behavioral Research*, 31, 1, 1-6.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

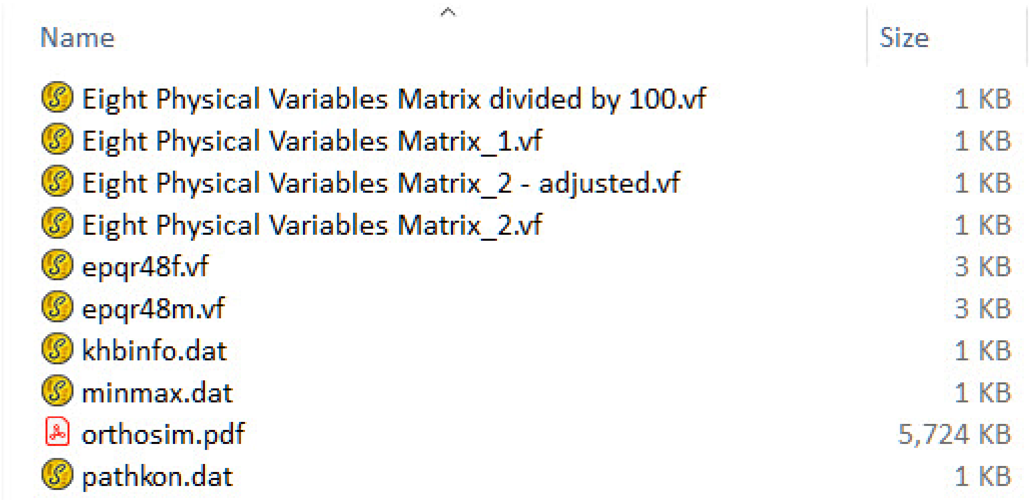
1.4 Installation details

Two program files are installed in the default 32-bit Windows Application subdirectory:


C:\Program Files (x86)\Orthosim
File
Home
Share
View
This PC › OS (C:) › Program Files (x86) › Orthosim
Quick access
Desktop
Downloads

Name	Size	Type
khb.exe	974 KB	Application
orthosim.exe	5,174 KB	Application

and example, program information files, and the pdf version of the program-accessed web-help (orthosim.pdf) are stored in: **C:\Users\Public\Orthosim**



Name	Size
Eight Physical Variables Matrix divided by 100.vf	1 KB
Eight Physical Variables Matrix_1.vf	1 KB
Eight Physical Variables Matrix_2 - adjusted.vf	1 KB
Eight Physical Variables Matrix_2.vf	1 KB
epqr48f.vf	3 KB
epqr48m.vf	3 KB
khbinfo.dat	1 KB
minmax.dat	1 KB
orthosim.pdf	5,724 KB
pathkon.dat	1 KB

The installation routine creates a program icon on the desktop:

Part II

2 Chapter 2: Similarity Indices

2.1 Why Four Similarity indices?

Because as Example #1 makes clear, neither the Pearson nor Congruence coefficient are sensitive to differences in magnitude between factor loadings or MDS coordinates. While mostly not a serious problem in factor analysis, coordinate scale differences (after origin centering, reflection, normalization, and rotation) and are critical to MDS - where the distance between coordinate values is itself indicative of similarity/dissimilarity.

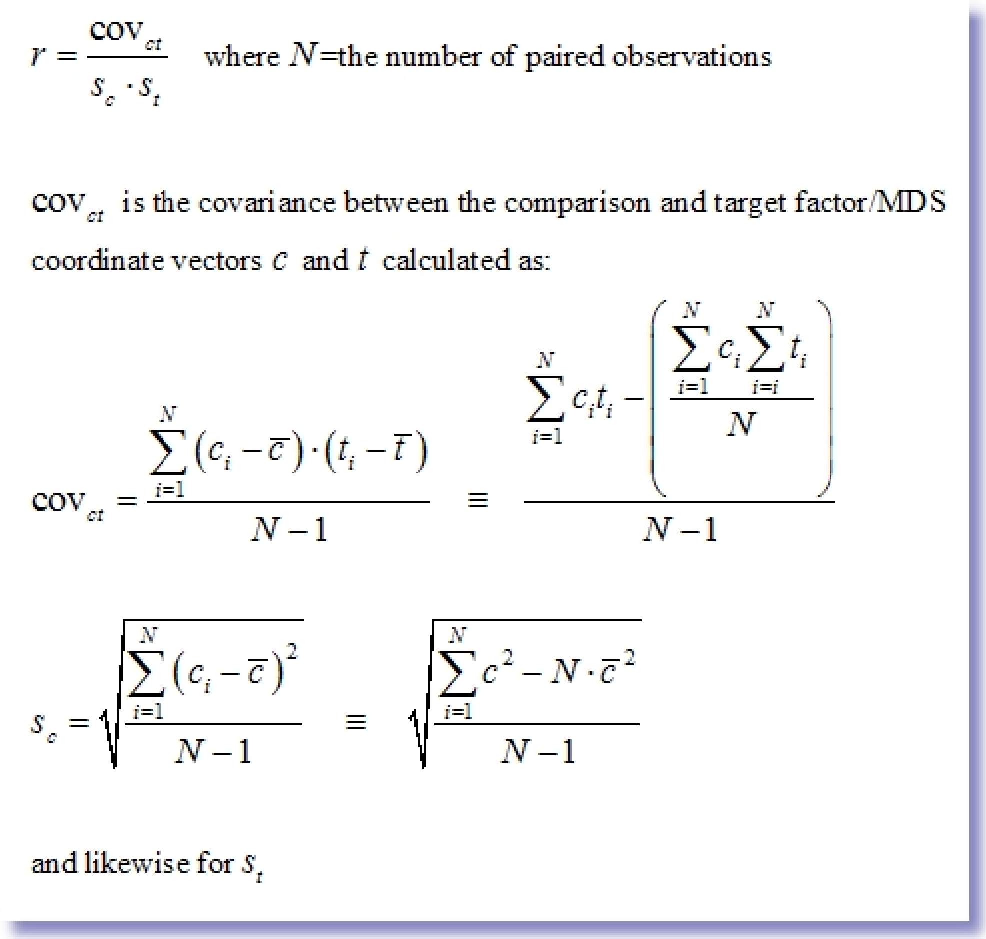
The problem is that the Pearson and Congruence coefficients are scale insensitive. That is, the coefficients are blind to the magnitude differences between loadings or coordinates, being sensitive to the monotonic order only.. So, we need to use a coefficient which is sensitive to both monotonicity relations and magnitude differences of corresponding loading/coordinate values. Two coefficients, designed by myself are used. A double-scaled euclidean distance coefficient expressed as a similarity measure, and a kernel-smoothed distance coefficient, also expressed as a similarity measure. Both coefficients vary between 0 and 1, where 0 = maximum possible discrepancy and 1 = absolute identity.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

2.2 The Pearson Coefficient

The normalised covariance between two variables. Here the variables are the factors or MDS coordinate "axes", and the loadings or coordinate values are considered the observations.So, the deviation-value (each variable observation is subtracted from the mean of the all the observations for that factor/axis) cross-products of two variables, divided by the product of the respective standard deviationstandard deviation is the second-moment around the mean). This is why it is sometimes called the Pearson product-moment correlation. The coefficient varies between +1.0 and -1.0, with +1.0 indicating maximum similarity, -1.0 = maximum inverse similarity, and 0.0 indicating no relationship whatsoever.

The formula is:



$r = \dfrac{\mathrm{cov}_{ct}}{s_c \cdot s_t}$ where N=the number of paired observations

cov_{ct} is the covariance between the comparison and target factor/MDS coordinate vectors c and t calculated as:

$$\mathrm{cov}_{ct} = \frac{\sum_{i=1}^{N}(c_i - \bar{c})\cdot(t_i - \bar{t})}{N-1} \equiv \frac{\sum_{i=1}^{N} c_i t_i - \left(\dfrac{\sum_{i=1}^{N} c_i \sum_{i=i}^{N} t_i}{N}\right)}{N-1}$$

$$s_c = \sqrt{\frac{\sum_{i=1}^{N}(c_i - \bar{c})^2}{N-1}} \equiv \sqrt{\frac{\sum_{i=1}^{N} c^2 - N\cdot\bar{c}^2}{N-1}}$$

and likewise for s_t

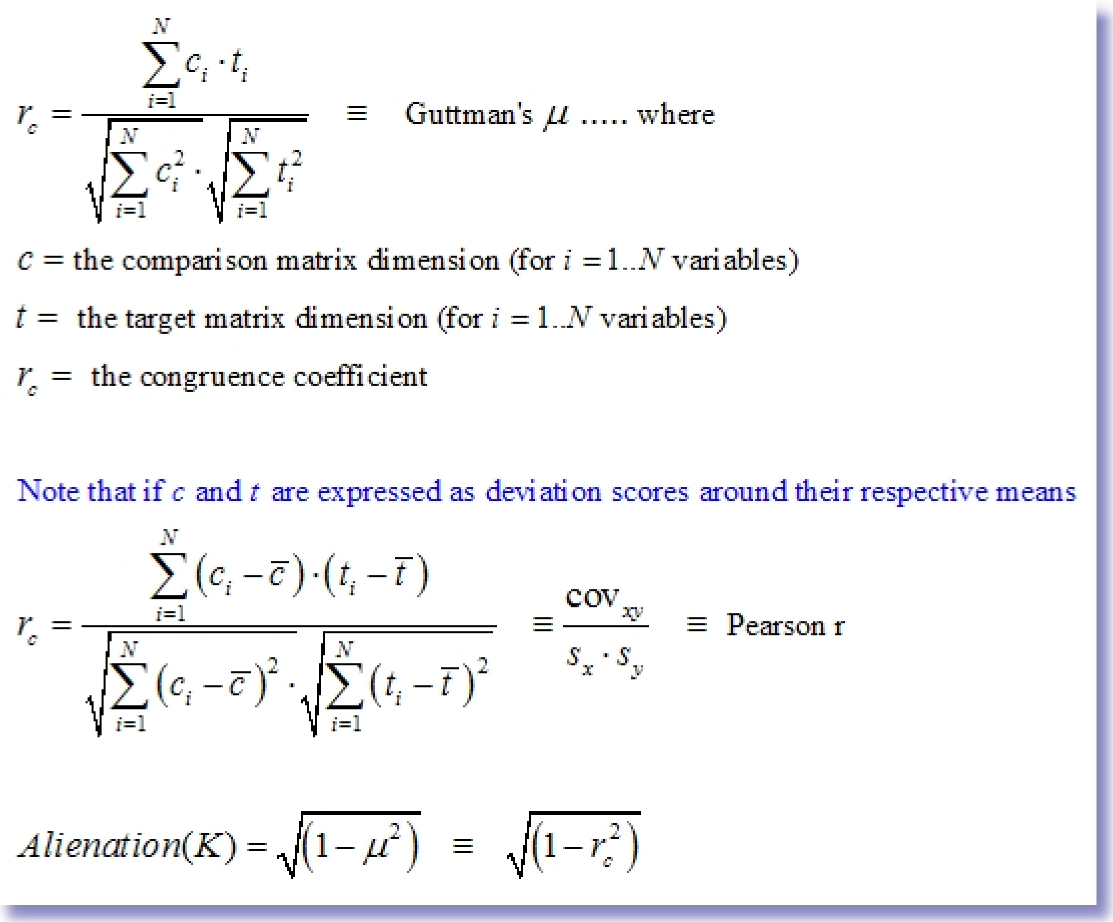
The problem with the Pearson correlation is that it is insensitive to the scale of the corresponding pairs of loadings or MDS coordinates, as Example 1 demonstrates. As such, it is not advisable to use this coefficient at all - but some investigators still like to see it all the same - so I have included it here for comparative purposes etc.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

2.3 The Congruence Coefficient

The congruence coefficient is normally encountered only in the domain of factor similarity (comparing factors from a factor analysis across samples). It is, in effect, a pearson correlation computed using unstandardised variables. That is, if we were to express each profile vector score as a deviation from the respective vector mean (subtract the elevation (mean) coefficient from each profile value), then compute the measure of relationship using the Pearson correlation formulae, we would indeed have computed a Pearson correlation coefficient. The congruence coefficient varies between +1.0 and -1.0, with +1.0 indicating maximum similarity, -1.0 = maximum inverse similarity, and 0.0 indicating no relationship whatsoever.

The formula is:



$$r_c = \frac{\sum_{i=1}^{N} c_i \cdot t_i}{\sqrt{\sum_{i=1}^{N} c_i^2} \cdot \sqrt{\sum_{i=1}^{N} t_i^2}} \quad \equiv \quad \text{Guttman's } \mu \text{ ..... where}$$

$c =$ the comparison matrix dimension (for $i = 1..N$ variables)

$t =$ the target matrix dimension (for $i = 1..N$ variables)

$r_c =$ the congruence coefficient

Note that if c and t are expressed as deviation scores around their respective means

$$r_c = \frac{\sum_{i=1}^{N} (c_i - \bar{c}) \cdot (t_i - \bar{t})}{\sqrt{\sum_{i=1}^{N} (c_i - \bar{c})^2} \cdot \sqrt{\sum_{i=1}^{N} (t_i - \bar{t})^2}} \equiv \frac{\text{cov}_{xy}}{s_x \cdot s_y} \equiv \text{Pearson r}$$

$$Alienation(K) = \sqrt{(1 - \mu^2)} \quad \equiv \quad \sqrt{(1 - r_c^2)}$$

Overall and Klett (1972, pp. 392-393) also refer to this coefficient as the **raw vector product coefficient** (contrasted with the normalized vector product coefficient - i.e. the Pearson correlation).

Guttman (1981) derived exactly the same coefficient for computing the similarity between derived and observed distances within multidimensional scaling (MDS); he called it the **monotonicity coefficient** (mu or m). Coxon (1982, p. 89-90) also shows that m is directly related to measures of stress (in multidimensional scaling), as well as Gutmman's **coefficient of Alienation (K)**. This alienation coefficient is a measure of the "unexplained" variation between the comparison and target profiles. So for example, of a congruence coefficient of 0.99 is computed, Guttman's K = 0.14. A brief discussion of these issues can be found in Borg and Groenen (1997, pp 203-204). In relation to this issue, they note that m takes on values close to 1.0 even if the MDS solution is "far from perfect". It was for this reason that Guttman converted m to K (the coefficient of Alienation), in order to expand the indicative range. If we subtract K from 1.0, we can obtain a measure of "similarity" (lack of alienation) that is based upon the expanded range of K.

Of interest is that Gorsuch (1983, p. 285), in relation to the use of the congruence coefficient for comparing factors, states "Occasionally, coefficients of congruence can give ridiculous results. Pinneau and Newhouse (1964) have pointed out that the index is highly influenced by the level and sign of loadings. **Factors whose loadings are the same size will, of necessity, have a high coefficient of congruence even if the patterns are unrelated**".

But, it has serious scale-sensitivity problems … so sensitive that I no longer recommend its usage at all.
The Pearson is based upon standardized deviations from the mean of each factor/coordinate axis loadings/values, while the congruence is based upon raw deviations from 0.0. What this means is that the Pearson is not sensitive to differences between the means of two sets of loadings, but the congruence might be expected to be. **In fact it is sensitive only when one vector contains loadings that vary proportionately around 0.0**. Jensen (1988) p. 100 gives a small example which shows the sensitivity of the congruence coefficient to the scale of the loadings/coordinate values.

Loadings/Coordinate values	Factor/Dimension 1	Factor/Dimension 2
1	.9	.4
2	.8	.3
3	.7	.2
4	.6	.1
5	.5	0.0
6	.4	-.1
7	.3	-.2
8	.2	-.3
9	.1	-.4

The Pearson r for these data is **1.00**, The congruence coefficient is **0.46.** But look at these data …

Loadings/Coordinate values	Factor/Dimension 1	Factor/Dimension 2
1	.9	.09
2	.8	.08
3	.7	.07
4	.6	.06
5	.5	.05
6	.4	.04
7	.3	.03
8	.2	.02
9	.1	.01

The Pearson r for these data is **1.00**, The congruence coefficient is also **1.00.** If we try the following data ....

Loadings/Coordinate values	Factor/Dimension 1	Factor/Dimension 2
1	.9	.09
2	.8	.08
3	.7	.07
4	.6	.06
5	.5	.0
6	.4	.0
7	.3	.0
8	.2	.0
9	.1	.0

The Pearson is **0.91** whilst the congruence is **0.90**. Yet 5 out of the 9 loadings on the 2nd dimension are 0, and the remainder less than 0.1 This is not "scale sensitivity", neither is the coefficient "sensitive" to the scale mean disparity between factors. In short, it's all but useless (like the Pearson correlation) for this kind of work - **unless** other information about the scale/magnitude differences is acquired. It is for this reason that I will no longer use the congruence coefficient any more unless its value is corroborated by using an index which is sensitive to loading/coordinate magnitudes as well as monotonic relations.i.e. Double-scaled euclidean or the kernel smoothed distance measures of agreement.

However, it is probably fair to say that most factor solutions which have previously used the congruence coefficient are probably Ok - in that the loadings are invariably of similar size and scale and so its value could be considered a fair representation of the agreement between two vectors, As I say, Orthosim has the capacity to advise on magnitude differences as well as monotonicity, it is as well to use this information.

References

Borg, I. and Groenen, P. (1997) Modern Multidimensional Scaling: Theory and Application. Springer Verlag. ISBN: 0-387-94865-7

Coxon, A.P.M. (1982) The User's Guide to Multidimensional Scaling. Heinemann Educational Books.ISBN: 0-435-82251-9

Gorsuch, R.L. (1983) Factor Analysis 2nd Edition. Lawrence Erlbaum. ISBN: 0-89859-202-X

Jensen, A.R. (1998) The g Factor: The Science of Mental Ability. Praeger Press. ISBN: 0-275-961-3-6

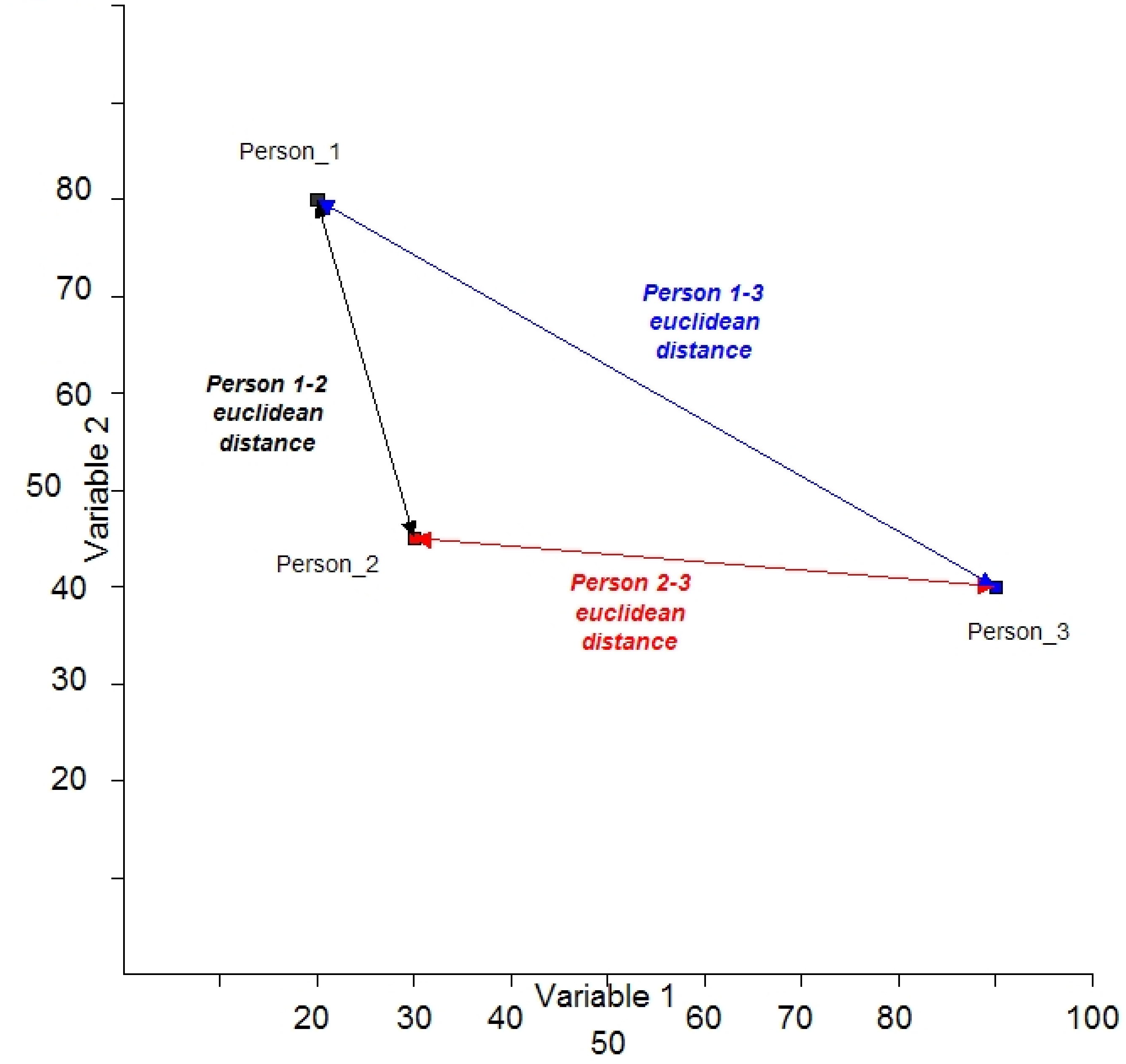
Overall, J.E. and Klett, C.J. (1972) *Applied Multivariate Analysis*. McGraw-Hill. ISBN: 07-047935-6

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

2.4 Double-Scaled Euclidean Distance

The **Euclidean metric** (and distance magnitude) is that which corresponds to everyday experience and perceptions. That is, the kind of 1, 2, and 3-Dimensional linear metric world where the distance between any two points in space corresponds to the length of a straight line drawn between them. The figure below shows the scores of three individuals on two variables (Variable 1 is the x-axis, Variable 2 the y-axis) –

Figure 1


Person_1
Person_2
Person_3
Person 1-2 euclidean distance
Person 1-3 euclidean distance
Person 2-3 euclidean distance
Variable 2
80
70
60
50
40
30
20
Variable 1
20
30
40
50
60
70
80
100

The straight line between each "Person" is the Euclidean distance. There would this be three such distances to compute, one for each person-to-person distance. Likewise, when calculating the distance between variables located on the same factor/MDS coordinate axis across two studies ...

The formula for calculating the distance between each of the three individuals as shown above is:

$$d = \sqrt{\sum_{i=1}^{v} (p_{1i} - p_{2i})^2}$$

where the difference between two persons' scores is taken, and squared, and summed for v variables (in our example v=2). Three such distances would be calculated, for p1 – p2, p1 – p3, and p2 - p3. In terms of comparison and target vector operations, the formula looks like ...

$$d = \sqrt{\sum_{i=1}^{N} (c_i - t_i)^2}$$

where we compare the loadings/MDS coordinate values (i=1 to N such values) across a factor/coordinate axis within the comparison and target data.

Normalised Euclidean Distance

The problem with the raw distance coefficient is that it has no obvious bound value for the maximum distance, merely one that says 0 = absolute identity. Its range of values vary from 0 (absolute identity) to some maximum possible discrepancy value which remains unknown until specifically computed. Raw Euclidean distance varies as a function of the magnitudes of the observations. Basically, you don't know from its size whether a coefficient indicates a small or large distance. The fact of the matter is that unless we know the maximum possible values for a Euclidean distance, we can do little more than rank dissimilarities, without ever knowing whether any or them are actually similar or not to one another in any absolute sense. A further problem is that raw Euclidean distance is sensitive to the scaling of each constituent variable. For example, comparing persons across variables whose score ranges are dramatically different. Likewise, when developing a matrix of Euclidean coefficients by comparing multiple variables to one another, and where those variables' magnitude ranges are quite different. Several approaches are taken in an attempt to normalize the coefficient. These are described with examples in my Technical Whitepaper #6 entitled: "Euclidean Distance: raw, normalized, and double-scaled coefficients". None of these usual normalisation versions are seen as particularly useful. Note also that so far, all we can talk about is distance or dissimilarity, not similarity.

Double-Scaled Euclidean Distance (DSED)

When comparing two variables/persons, what we can do is to calculate the Euclidean distance from data which is transformed into a 0-1 metric using a strictly linear method (rather than non-linear normalization-standardization), then re-scale the resultant Euclidean distance measure itself into a 0-1 range, scaling it into a range defined by 0 through to the maximum possible distance observable between the two variables/persons. This also allows us to re-express this coefficient as a similarity measure by simply subtracting it from 1.0.

There are three computational steps to be followed in factor/MDS coordinate similarity with a DSED coefficient...(*in fact all of these are automated within the program - including the use of fixed maximum variable discrepancies*).

Step 1: Determine the maximum possible squared discrepancy for each factor/MDS coordinate dimension using the minimum and maximum values which might be observed for these factors/MDS coordinate axes. Call these values *md*.

Given a factor solution invariably presented to Orthosim will be based upon an input correlation matrix, all factor loadings are thus standardized between -1.0 and +1.0, hence we calculate the maximum possible discrepancy for any two variable loadings as:

$$md = (\text{min-max})^2$$

$$md = ((-1)-(+1))^2$$

$$md = 4$$

* Note that *md* acts as a constant.

Although MDS solution coordinates invariably possess a range larger than -1 to +1, the centering, reflection, and normalizing of the coordinates into the same metric space mean that the minimum and maximum values always range between -1 to +1, thus the resulting scaling of the squared disparity between any two coordinate values will be fixed at this range.

An important point: Orthosim only permits one minimum and maximum value (-1 to +1) to be used for **all** factor/MDS coordinate comparisons. This differs from the use of this coefficient in person-target profiling (as explained in my Technical Whitepaper #6 entitled: "Euclidean Distance: raw, normalized, and double-scaled coefficients".

Step 2. Compute the sum of squared discrepancies per observation, dividing through the squared discrepancy for each pair of observations by the maximum possible discrepancy observable given these two variables. Then take the square root of the sum to produce the scaled variable Euclidean distance.

we modify the equation above ...

$$d = \sqrt{\sum_{i=1}^{N} (c_i - t_i)^2}$$

to become ...

$$d_1 = \sqrt{\sum_{i=1}^{N} \left(\frac{(c_i - t_i)^2}{md} \right)}$$

where d1 = the "scaled variable" Euclidean distance
md = the maximum possible squared discrepancy between these two variables (**which is always 4 for each variable**).

Step 3. Compute the scaled value from step 3 by dividing it by the square root of N, where N = the number of paired loadings/MDS coordinate values.

$$d_2 = \frac{\sqrt{\sum_{i=1}^{N}\left(\frac{(c_i - t_i)^2}{md}\right)}}{\sqrt{N}} = \frac{d_1}{\sqrt{N}}$$

A worked example of this calculation is given in in my Technical Whitepaper #6 entitled: "Euclidean Distance: raw, normalized, and double-scaled coefficients".

Finally, we express this scaled distance (which varies between 0 = absolute identity to 1.0 = maximum possible dissimilarity) as a similarity for our purposes here, by subtracting it from 1.0. So, the formula above becomes:

$$s_2 = 1.0 - \left(\frac{\sqrt{\sum_{i=1}^{N}\left(\frac{(c_i - t_i)^2}{md}\right)}}{\sqrt{N}}\right) = 1.0 - \left(\frac{d_1}{\sqrt{N}}\right)$$

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

2.5 Kernel Smoothed Distance

My design of this coefficient was based upon the idea that a distance function could be ***shaped*** in such a way that if the simple arithmetic unsigned difference between a person's attribute value and the target attribute value was computed to be within a certain range, then the computed distance should reflect a very small distance, almost regardless of the actual distance. But, as that distance grew larger, then the computed distance should be accelerated in size. In short, an "inertial" effect was aimed for – translated into a distance metric.

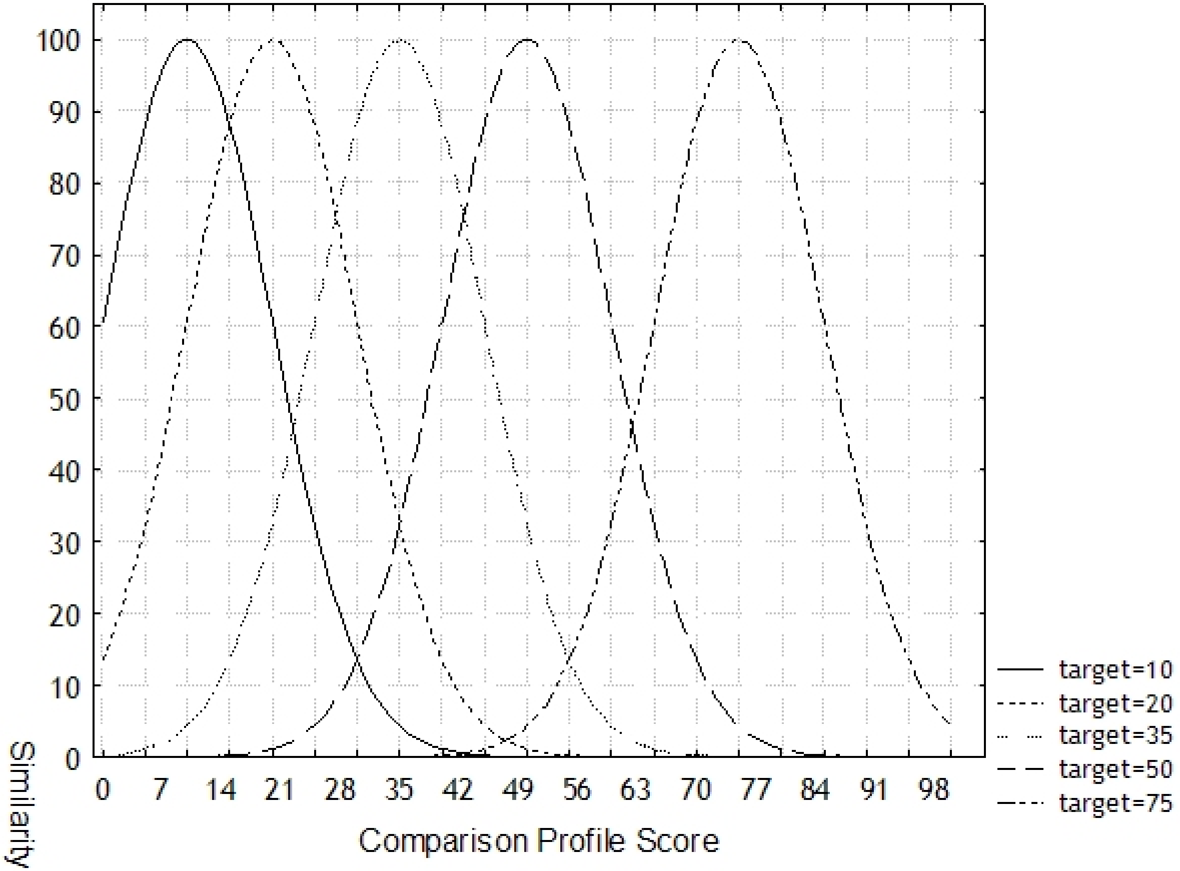
The "shaped distance" design is in direct response to a need to "take control" or "shape" the distance function between two objects in space, as a function of their distance. So, small disparities between two profile scores would have little effect on the distance calculation, but, as the disparity grew greater, so would the distance be degraded in an accelerated fashion. The function chosen to achieve this was the normal distribution curve equation, with distance degradation acceleration controlled by the standard deviation parameter and the target profile value represented by the mean of the distribution function.

Within the area of computational data smoothing and trend analysis, this kind of "inertial" effect is known as kernel density smoothing (Hastie et al, 2001). It has been applied here to permit the control of a distance function rather than to smooth data trends. As the distance between a target score value and a comparison score value increases, so the distance is accelerated between them once the distance moves beyond a region of little change (near the peak of such a distribution). The parameter controlling the "plateau" effect is the standard deviation of such a distribution; likewise, the acceleration in distance at larger values.

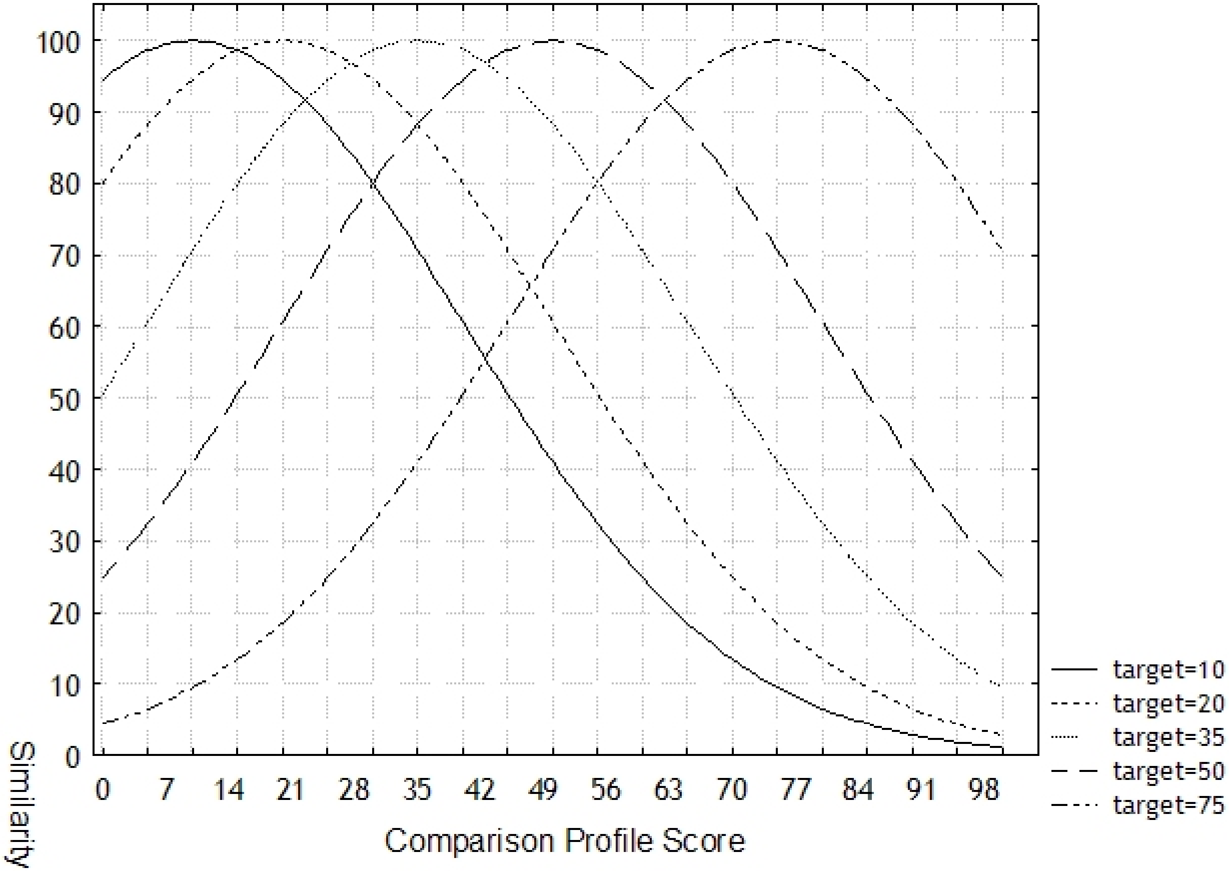
It has been applied here to permit the control of a distance function rather than to smooth data trends. As the distance between a target variable value and a comparison variable value increases, so the distance is "accelerated" (made much larger than it actually is) between them once the distance moves beyond a region of little change (near the peak of such a distribution).

The parameter controlling the "plateau" effect is the standard deviation of such a distribution; likewise, the acceleration in distance at larger values. For example, the figures below show the distance curves with standard deviations of 10 and 30 for five different target values of 10, 20, 35, 50, and 75. The curves are constructed by systematically varying comparative score values (those score values to be compared to the fixed target) from 0 through 100 in 1-unit steps.

A kernel distance weighting function with standard deviation = 10. Comparison scores ranging between 0 and 100 are compared to five target scores of 5, 10, 35, 50, and 75



A kernel distance weighting function with standard deviation = 30. Comparison scores ranging between 0 and 100 are compared to five target scores of 5, 10, 35, 50, and 75.



The formula for this coefficient is:

$$KSD = \frac{\sum_{i=1}^{N}\left[\frac{1}{s\sqrt{2\pi}}e^{-\left[\frac{(c_i - t_i)^2}{2s^2}\right]}\right]\cdot\left(100\cdot\left(s\cdot\sqrt{2\pi}\right)\right)}{N}$$

where *s* = standard deviation "smoothing" parameter

c = the comparison vector (factor/MDS coordinate axis)

t = the target vector (factor/MDS coordinate axis)

N = the number of variables being compared within a factor/MDS coordinate axis

It is constructed to yield a percentage similarity measure, with 0% indicating maximum dissimilarity and 100% being absolute identity.

In Orthosim, I simply divide it by 100 so as to express similarity in a unified metric of 0 to 1.

More details concerning this coefficient are provided in the chapter:

Barrett, P.T. (2005) Person-Target Profiling. In André Beauducel, Bernhard Biehl, Michael Bosniak, Wolfgang Conrad, Gisela Schönberger, and Dietrich Wagener (Eds.) *Multivariate Research Strategies: a Festschrift for Werner Wittman*. Chapter 4, pp 63-118. Hofstede and Huber.

Also, details of the kernel smoothing techniques are provided in:
Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

2.6 Root-Mean-Square Deviation

This coefficient is calculated as:

$$rms = \sqrt{\sum_{i=1}^{N}\left(\frac{(c_i - t_i)^2}{N}\right)} = \sqrt{\frac{\sum_{i=1}^{N}(c_i - t_i)^2}{N}}$$

where N = the number of variables and *c* and *t* are the comparison and target vectors respectively

It has no upper limit - being dependent upon the scale of the variables ... however, it is useful as a "relative change" coefficient which informs you as to how well (and the degree of improvement from the original input comparison matrix) the optimally rotated comparison matrix fits the target matrix.

Note that this coefficient is built from the simple least-squares difference expression used to "locate" the optimal position of the comparison matrix to the target matrix.

Paul Barrett, 30th December, 2019,   email: paul@pbarrett.net

Part III

3 Chapter 3: The Programs

3.1 The precise mechanics of Orthosim

Rather than try and present the entire suite of equations here, it is better you download the paper: Barrett, P.T., Petrides, K.V., Eysenck, S.B.G., and Eysenck, H.J. (1998) The Eysenck Personality Questionnaire: an examination of the factorial similarity of P, E, N, and L across 34 countries *Personality and Individual Differences,* 25, 5, 805-819. This provides a listing of all matrix operations used to perform the Kaiser-Hunka-Bianchini (KHB) congruential, least-squares optimised, rotation.

This particular (KHB) technique of factor comparison compares two row-normalized orthogonal matrices to one another. A comparison set of orthogonal factor vectors are rotated rigidly against a target set of vectors, such that the overall least-squares discrepancy between loadings/coordinate values in the comparison and target matrices is at a minimum. Even if the two matrices to be compared have been initially rotated to an oblique position, the Kaiser-Hunka-Bianchini (1971) procedure "undoes" the oblique transformations and performs a subsequent orthogonal rotation of one set of row-normalized vectors against the other. Thus, the KHB procedure is essentially an orthogonal procrustes procedure.

Unfortunately, as pointed out by Bijnen, Poortinga, and Ten Berge (1996), the original KHB similarity coefficients were mis-specified by Kaiser et al, who advocated using the angular shift cosines (required to bring the two factor matrices into maximum congruity) as measures of similarity. However, Barrett et al (1998) modified the procedure to permit the calculation of congruence coefficients over the maximally congruent orthogonalized factor matrices.

A key feature within Orthosim is that of row normalization of **factor** matrices to be compared prior to the comparison procedure being applied., and then the re-expression (reverse normalization) of the maximally congruent comparison matrix back in the metric of the original input matrix. These operations produce a slightly different final comparison matrix than if we just rotate the input comparison matrix "as is" (no row normalization selected). Further, that expansion/contraction (the procrustes effect) completely removes the scale differences between matrices (see Example 1). You must decide whether you wish to row-normalize or not.

Look at the following results from comparing two matrices via Orthosim - the 8 physical variable matrix given in Table 15.8 (p. 355) of Harman (1976) ... (the two matrices are included in the installation package for your convenience) ...

From Orthosim (with row normalization) - *with the factors re-expressed in their original metric space*

```
   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 1

            FAC. 1    FAC. 2
            ------    ------
Var_1       0.8712    0.2805
Var_2       0.9199    0.2067
Var_3       0.8867    0.1841
Var_4       0.8629    0.2508
Var_5       0.2303    0.9141
Var_6       0.1865    0.7781
Var_7       0.1268    0.7535
Var_8       0.2557    0.6529


 HYP.COUNT       0         0
 VARIANCE    3.3053    2.6532
```

From Orthosim (***<u>without</u>*** row normalization)

```
   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 1

            FAC. 1    FAC. 2
           ------     ------
Var_1       0.8763    0.2643
Var_2       0.9235    0.1896
Var_3       0.8900    0.1677
Var_4       0.8674    0.2348
Var_5       0.2471    0.9096
Var_6       0.2009    0.7745
Var_7       0.1407    0.7510
Var_8       0.2677    0.6481

 HYP.COUNT       0         0
 VARIANCE    3.3580    2.6005
```

Note that the eigenvalue sum remains constant between the two solutions- but that the actual eigenvalues for each factor vary marginally.

This also affects the overall similarity and factor/coordinate axis similarity measures:

From Orthosim (with row normalization)

Mean Solution Congruence = **0.96580**

```
                  Congruence Coefficients:  between the target and maximally congruent comparison matrix
                  **************************************************************************************


          FAC. 1    FAC. 2
          ------    ------
FAC. 1    0.9651    0.2524
FAC. 2    0.2516    0.9665
```

From Orthosim (*without* row normalization)

Overall Solution Congruence = **0.96563**

```
                  Congruence Coefficients:  between the target and maximally congruent comparison matrix
                  **************************************************************************************


          FAC. 1    FAC. 2
          ------    ------
FAC. 1    0.9706    0.2655
FAC. 2    0.2337    0.9560
```

References

Barrett, P.T., Petrides, K.V., Eysenck, S.B.G., and Eysenck, H.J. (1998) The Eysenck Personality Questionnaire: an examination of the factorial similarity of P, E, N, and L across 34 countries. *Personality and Individual Differences,* 25, 5, 805-819

Bijnen, E.J. (1988) The Questionable value of cross-cultural comparisons with the Eysenck Personality Questionnaire. *Journal of Cross-Cultural Psychology*, 19, 2, 193-202.

Bijnen, E.J., van der Net, T.Z.J., Poortinga, Y.H. (1986) On cross-cultural comparative studies with the Eysenck Personality Questionnaire. *Journal of Cross-Cultural Psychology*, 17, 1, 3-16.

Harman, H. (1976) Modern Factor Analysis, 3rd edition. Chicago University Press. ISBN: 0-226-31652-1

Kaiser, H.F., Hunka, S., and Bianchini, J.C. (1971) Relating factors between studies based upon different individuals. *Multivariate Behavioral Research*, 6, 409-422.

Ten Berge, J.M.F. (1996) The Kaiser, Hunka, and Bianchini factor similarity coefficients: a cautionary note. *Multivariate Behavioral Research*, 31, 1, 1-6.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

3.2 Job Control and Input Data File Structures

Orthosim_3 is written in Delphi 2010, but the routines used by the program are written in Fortran 77 and compiled using Approximatrix Simply Fortran v3.7. However, because my original program (19989/1992) was job-control-based as were all Fortran programs around that time, I have had to continue the convention until I can completely re-write the software in Delphi.

Anyway, the upshot of this is that the input files possess a fixed format structure, and must appear as "text files" to the program. This is a real pain.

For those using SPSS or whatever, or just wanting to set up a file from say a paper or book chapter, or even set up a synthetic target file (1s and 0s as target loadings say), you need to be aware of how to do this.

A Synthetic Target

It is possible to set up a synthetic target data matrix and diagonal factor correlation matrix (1s in the diagonal and 0s elsewhere). You could use my free-to-download utility program: Target Matrix Generator, or just create one yourself.

The dummy matrix would itself consist of 1s and 0s - and provide a clear target pattern for comparison purposes. For example, take the following matrix which has been formed from an expectation that the first 5 variables would load factor 1, and the second 5 on factor 2. The matrix would look like:

F1	**F2**
1.0	0.0
1.0	0.0
1.0	0.0
1.0	0.0
1.0	0.0
0.0	**1.0**
0.0	**1.0**
0.0	**1.0**
0.0	**1.0**
0.0	**1.0**

with the orthogonal factor correlation matrix as

1.0	0.0
0.0	1.0

The next section details the Orthosim input file structure - common to both routines so you can submit either file setup to either program.

Some "runtime" specifics

When running the windows program Orthosim_3, it creates a temporary file in the public subdirectory: **C:\Users\Public\Orthosim**, called "khbinfo.dat" which contains the output "length" instructions, minimum, maximum, and KSD smoothing values appropriate for the orthosim routine, along with the file paths and names of the input and output files you selected. The fortran routine (khb.exe in **C:\Program Files (x86)\Orthosim**) is then run by Orthosim, and this routine reads the appropriate temporary file information and 'does the business'. However, for the user, it's all seamless and entirely automated.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

This is the map for the job control that heads an input file for Orthosim coordinate comparison analysis. A throwback from the 1990s and Fortran 77! The input file should contain ...

LINE 1 .... TITLE (up to 80 chars)

LINE 2 .... Col. 1-2 the number of factors in the matrix

Col. 3-5 the number of variables/items in the matrix.

NEXT LINE/s ... the names of the variables, 8 chars per name, with up to 10 names per line. The input format is fixed at (10A8)

NEXT LINE ..... the INPUT data format for the factor PATTERN matrix

NEXT LINE ..... the INPUT data format for the factor INTERCORRELATION matrix.

An example file setup , for 3 factors and 5 variables, looks like ...

```
Ahmavaara Test Matrix -X-
03005
Variab_1Variab_2Variab_3Variab_4Variab_5
(8F6.3)
(8F6.3)
   .56   .43   .31   .36   .35
   .12  -.03   .32   .09  -.03
   .24   .50   .24   .31   .55
  1.00   .00   .00
   .00  1.00   .00
   .00   .00  1.00
```

Note that the factor correlation matrix in this example indicates an orthogonal solution. Also, the factor correlation matrix always follows the factor pattern matrix. Note also that each factor/MDS coordinate axis is listed "by row". That is, the loadings/coordinate values for each dimension are listed factor/axis by factor/axis.

The 8 physical variable comparison matrix given in Table 15.8 (p. 355) of Harman (1976) looks like

```
Eight Physical Variables - Matrix No. 1
 2  8
Var_1   Var_2   Var_3   Var_4   Var_5   Var_6   Var_7   Var_8
(8F12.6)
(8F6.3)
        .856        .848        .808        .831        .750        .631       .569         .607
       -.324       -.412       -.409       -.342        .571        .492       .510         .351
  1.000  0.000
  0.000  1.000
```

Another example – using a sample questionnaire oblique factor matrix input – with 15 items and 3 factors …

```
Questionnaire Data - AoHT - items 12 and 14 excluded - 3 factors DIROBL
 3 15
AoHT 1  AoHT 2  AoHT 3  AoHT 4  AoHT 5  AoHT 6  AoHT 7  AoHT 8  AoHT 9  AoHT 10
AoHT 11 AoHT 13 AoHT 15 AoHT 16 AoHT 17
(8F10.6)
(8F10.6)
   .729944   .595202   .778294   .680963  -.097802   .034107   .244660   .391214
   .360160   .387810   .042117  -.139163   .606636   .094208   .409074
   .217157   .040858  -.151194  -.141103  -.888941  -.911463  -.036675  -.242526
  -.224271  -.508952  -.009535  -.052023  -.032373  -.703377  -.309347
  -.008865  -.319628   .218008   .003275  -.048440  -.001285  -.747034  -.156402
  -.282549  -.075753  -.839402  -.780177  -.045437  -.053086  -.150631
  1.000000  -.309750  -.264361
  -.309750  1.000000   .231690
  -.264361   .231690  1.000000
```

Yes - you've probably also noticed the Fortran format statements ..8F10.6 etc! This means a fixed field of 10 characters, 8 per line, and 6 decimal places per number.
8F6.3 = a fixed field of 6 characters, 8 per line, and 3 decimal places per number.
108F12.8 = a fixed field of 12 characters, 10 per line, and 8 decimal places per number.
16F12.6 = a fixed field of 12 characters, 16 per line, and 6 decimal places per number.

Reference

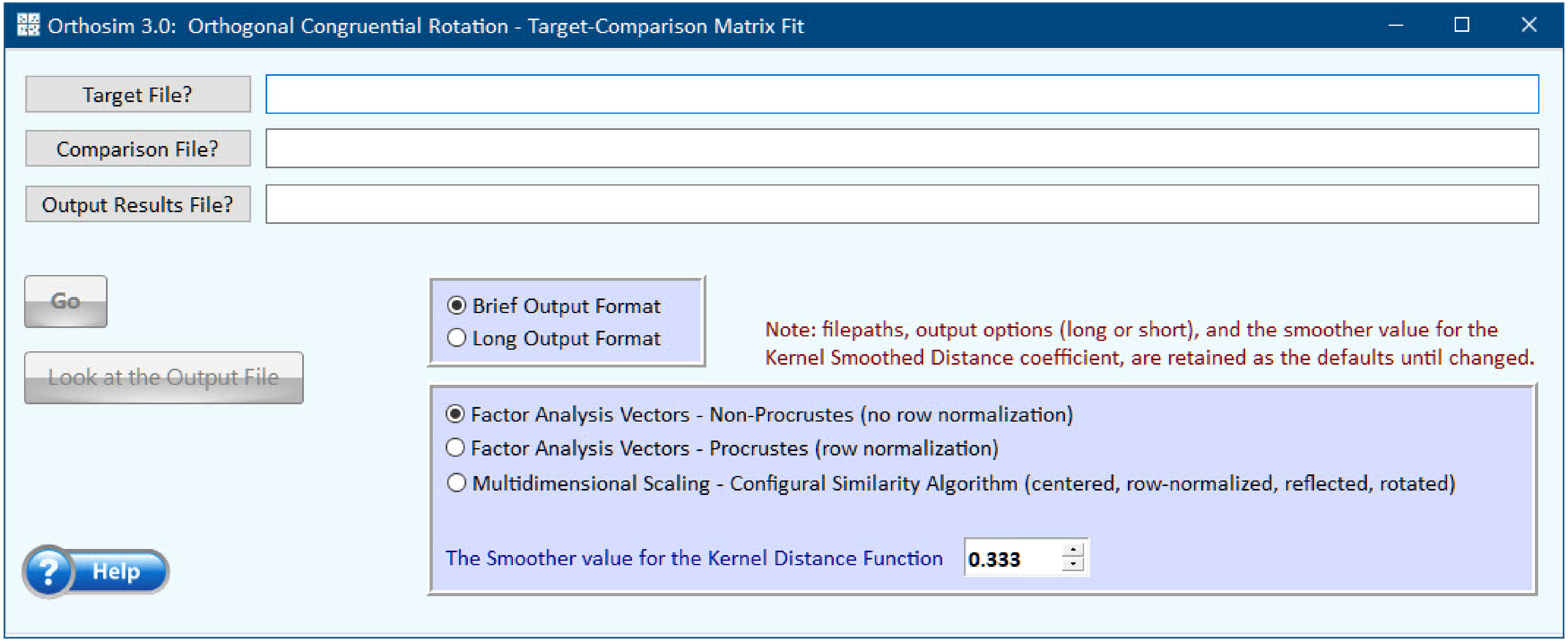
Harman, H. (1976) Modern Factor Analysis, 3rd edition. Chicago University Press. ISBN: 0-226-31652-1

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

3.3 How to Run an Analysis

Click on the program icon on your desktop:

The following screen will appear - which is the orthosim 3 user interface ...



Orthosim 3.0: Orthogonal Congruential Rotation - Target-Comparison Matrix Fit

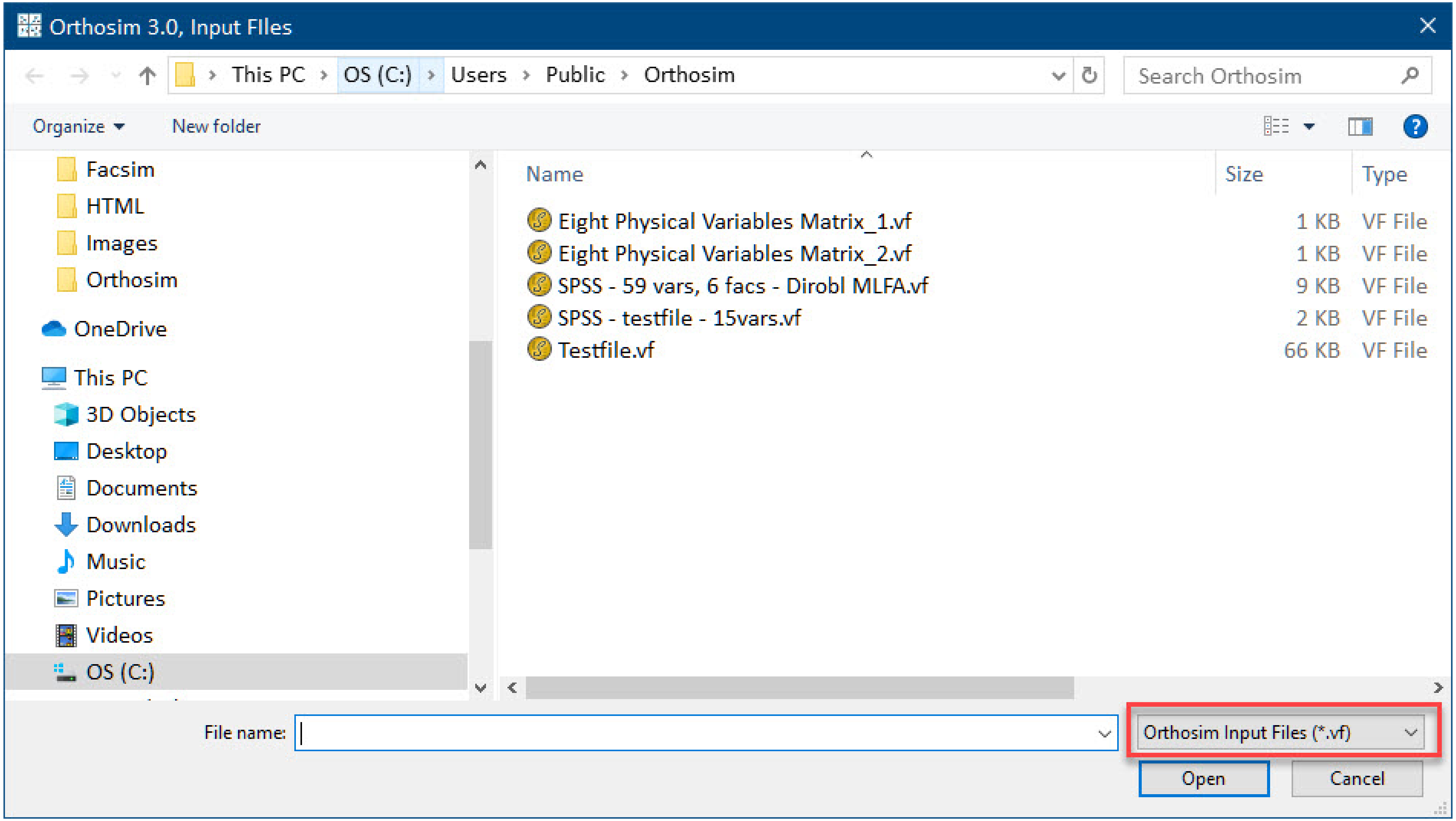
● Brief Output Format
○ Long Output Format

Note: filepaths, output options (long or short), and the smoother value for the Kernel Smoothed Distance coefficient, are retained as the defaults until changed.

● Factor Analysis Vectors - Non-Procrustes (no row normalization)
○ Factor Analysis Vectors - Procrustes (row normalization)
○ Multidimensional Scaling - Configural Similarity Algorithm (centered, row-normalized, reflected, rotated)

The Smoother value for the Kernel Distance Function 0.333

Step 1: Select a target matrix file ... by clicking on the "Target File?" button. This will display a dialog screen where you can select your file ...

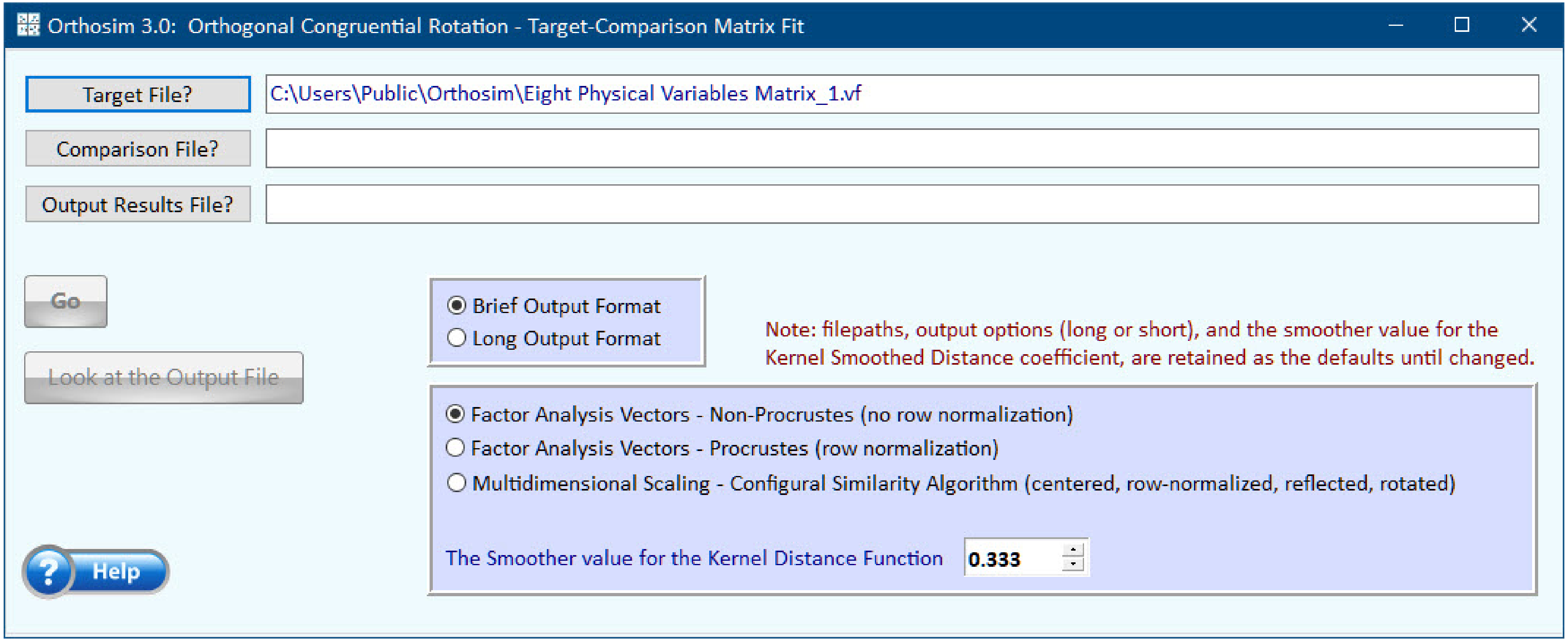


Name	Size	Type
Eight Physical Variables Matrix_1.vf	1 KB	VF File
Eight Physical Variables Matrix_2.vf	1 KB	VF File
SPSS - 59 vars, 6 facs - Dirobl MLFA.vf	9 KB	VF File
SPSS - testfile - 15vars.vf	2 KB	VF File
Testfile.vf	66 KB	VF File

Orthosim Input Files (*.vf)

You'll notice that you are only seeing files which end in ".vf". This is the "Facsim/Orthosim" default suffix for an input file. If you want to see "all Files", just click on the highlighted box.

Select your file ...



Orthosim 3.0: Orthogonal Congruential Rotation - Target-Comparison Matrix Fit

Target File? C:\Users\Public\Orthosim\Eight Physical Variables Matrix_1.vf

Comparison File?

Output Results File?

Go

Look at the Output File

● Brief Output Format
○ Long Output Format

Note: filepaths, output options (long or short), and the smoother value for the Kernel Smoothed Distance coefficient, are retained as the defaults until changed.

● Factor Analysis Vectors - Non-Procrustes (no row normalization)
○ Factor Analysis Vectors - Procrustes (row normalization)
○ Multidimensional Scaling - Configural Similarity Algorithm (centered, row-normalized, reflected, rotated)

The Smoother value for the Kernel Distance Function 0.333

You now see the filename in the row next to the Target File button. Do the same for the "Comparison File" and the "Output Results File". You will note that the output results file is defaulted to **sim1.txt** ... you can change this at any time.

As you enter the Output results filename, you will notice the "Go!" button is activated ... which means you can now implement an analysis ...

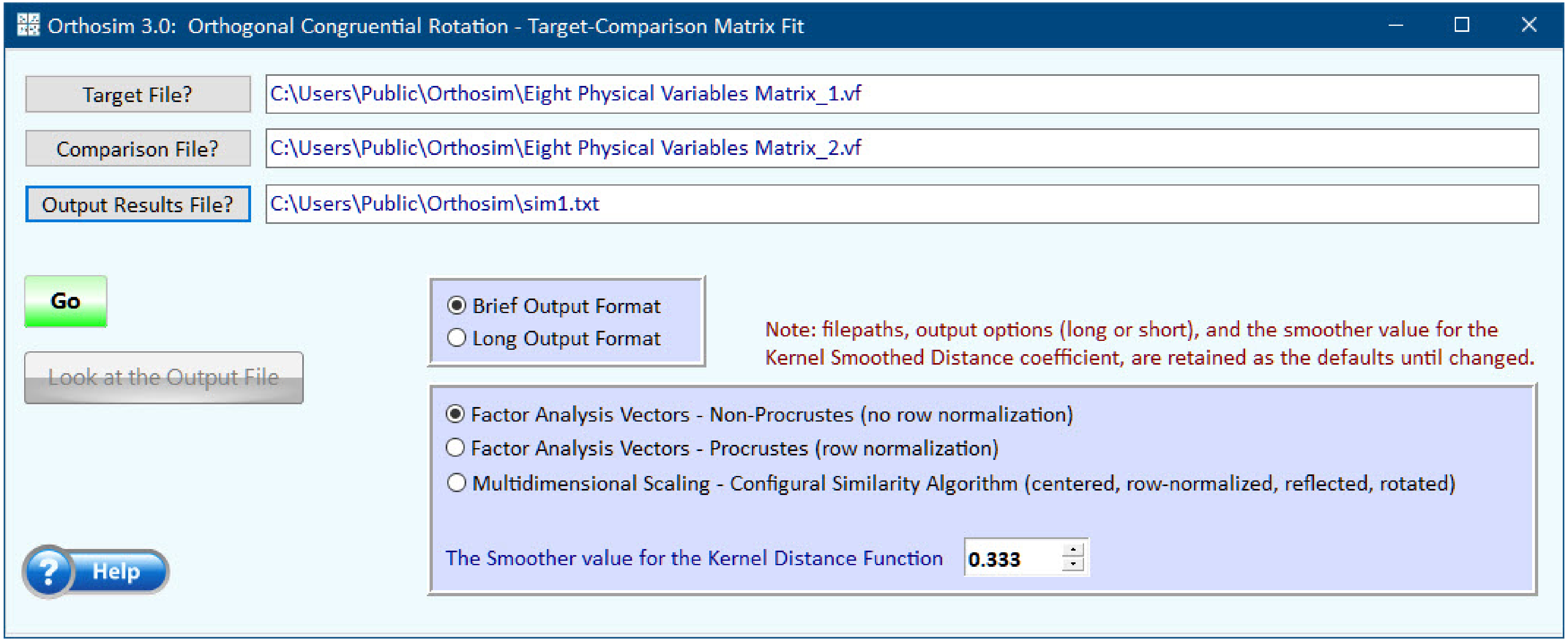
However ...before you hit the "Go" button - choose which kind of analysis you'd like to run, and whether you'd like brief or long output.

Finally, check that the KSD smoothing value is acceptable. It is set at a fairly sensitive "discrepancy degredation" level - so you might want to increase it say to 0.4. However, remember this is a coefficient you need to calibrate on some specially adjusted datasets, taking into account how you you will adjudge "similarity" and how sensitive you want this coefficient to be to any kind of discrepancy,

You'll note that the program "remembers" the directory you last used to input and save output files, as well as the last specified KSD smoothing coefficient value.

Ready? Then click on the button.

Almost instantaneously, the "Look at the Output File" buttons will be enabled - signaling the finish of the analysis.



Orthosim 3.0: Orthogonal Congruential Rotation - Target-Comparison Matrix Fit

Target File? C:\Users\Public\Orthosim\Eight Physical Variables Matrix_1.vf

Comparison File? C:\Users\Public\Orthosim\Eight Physical Variables Matrix_2.vf

Output Results File? C:\Users\Public\Orthosim\sim1.txt

● Brief Output Format
○ Long Output Format

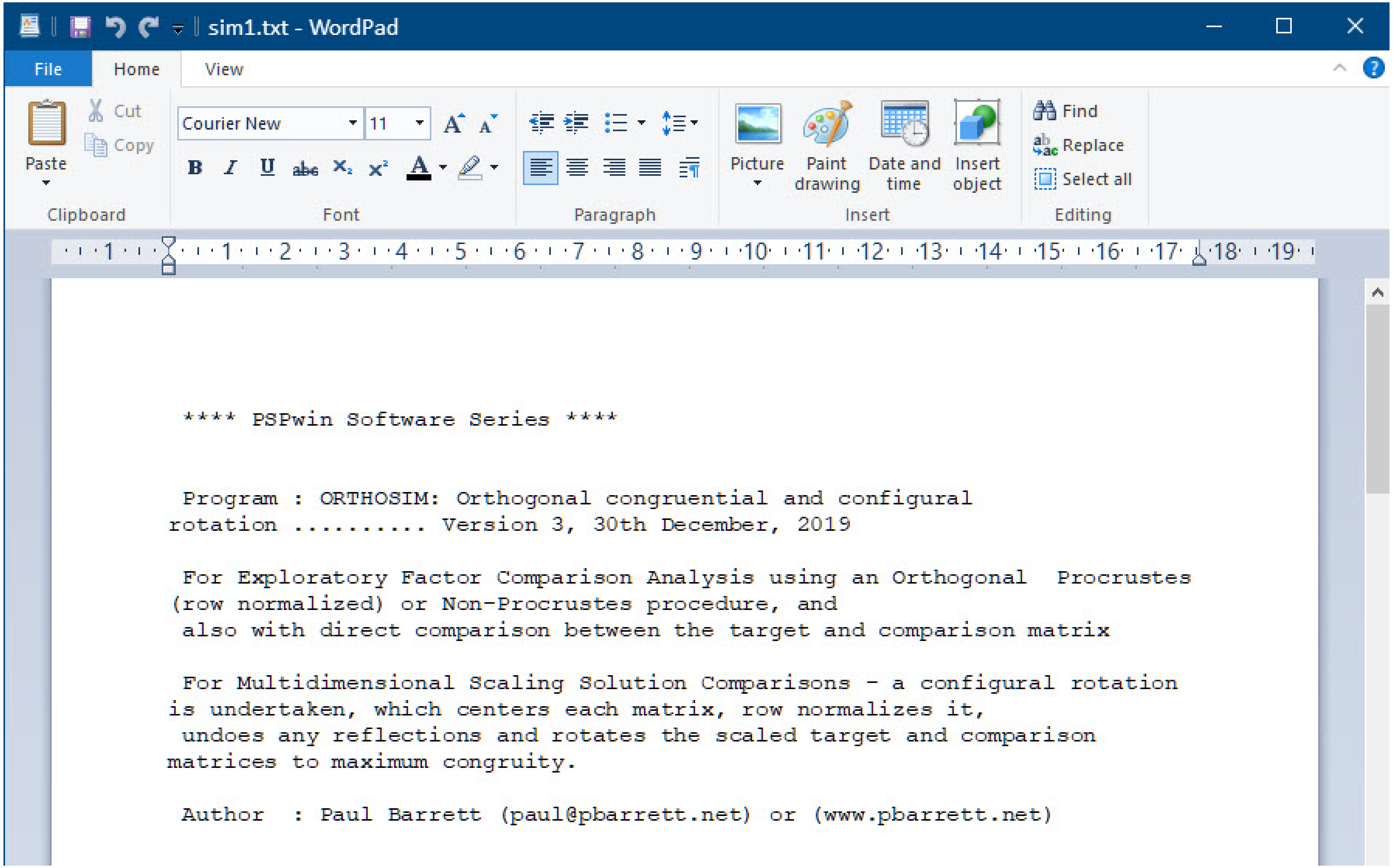
Note: filepaths, output options (long or short), and the smoother value for the Kernel Smoothed Distance coefficient, are retained as the defaults until changed.

● Factor Analysis Vectors - Non-Procrustes (no row normalization)
○ Factor Analysis Vectors - Procrustes (row normalization)
○ Multidimensional Scaling - Configural Similarity Algorithm (centered, row-normalized, reflected, rotated)

The Smoother value for the Kernel Distance Function 0.333

If you click on the "Look at Output File" button, your output file will be opened up using Windows **Wordpad** (which is a rich text-editor installed as part of all Windows 7 upwards installations).

Initially, the file will open as portrait format with wide default margins.

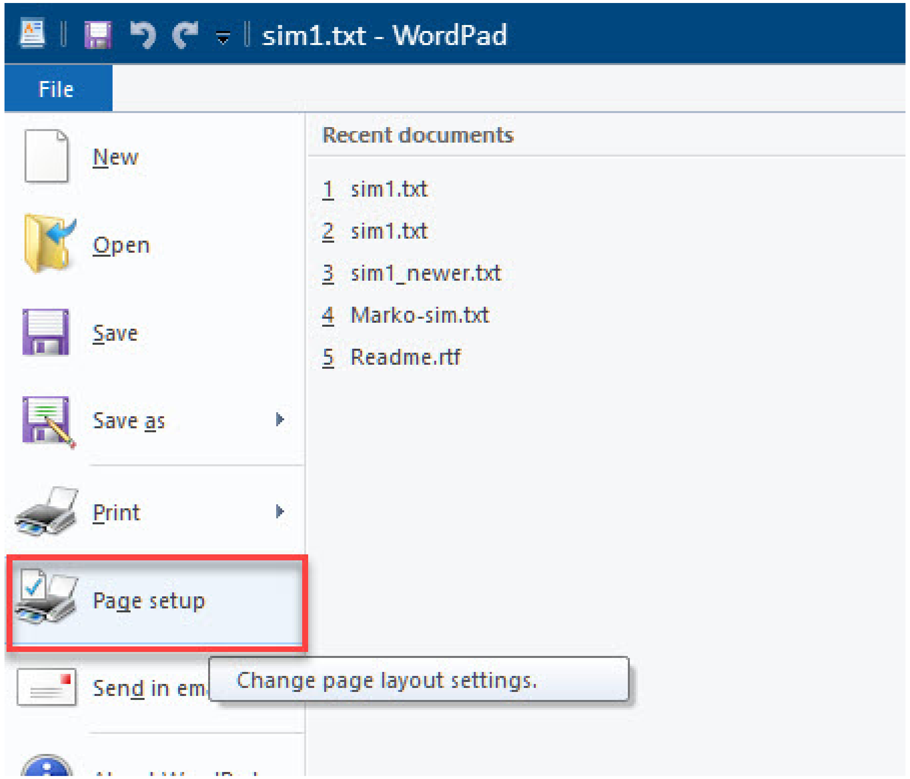


```
 **** PSPwin Software Series ****


 Program : ORTHOSIM: Orthogonal congruential and configural
rotation .......... Version 3, 30th December, 2019

 For Exploratory Factor Comparison Analysis using an Orthogonal  Procrustes
(row normalized) or Non-Procrustes procedure, and
 also with direct comparison between the target and comparison matrix

 For Multidimensional Scaling Solution Comparisons - a configural rotation
is undertaken, which centers each matrix, row normalizes it,
 undoes any reflections and rotates the scaled target and comparison
matrices to maximum congruity.

 Author  : Paul Barrett (paul@pbarrett.net) or (www.pbarrett.net)
```

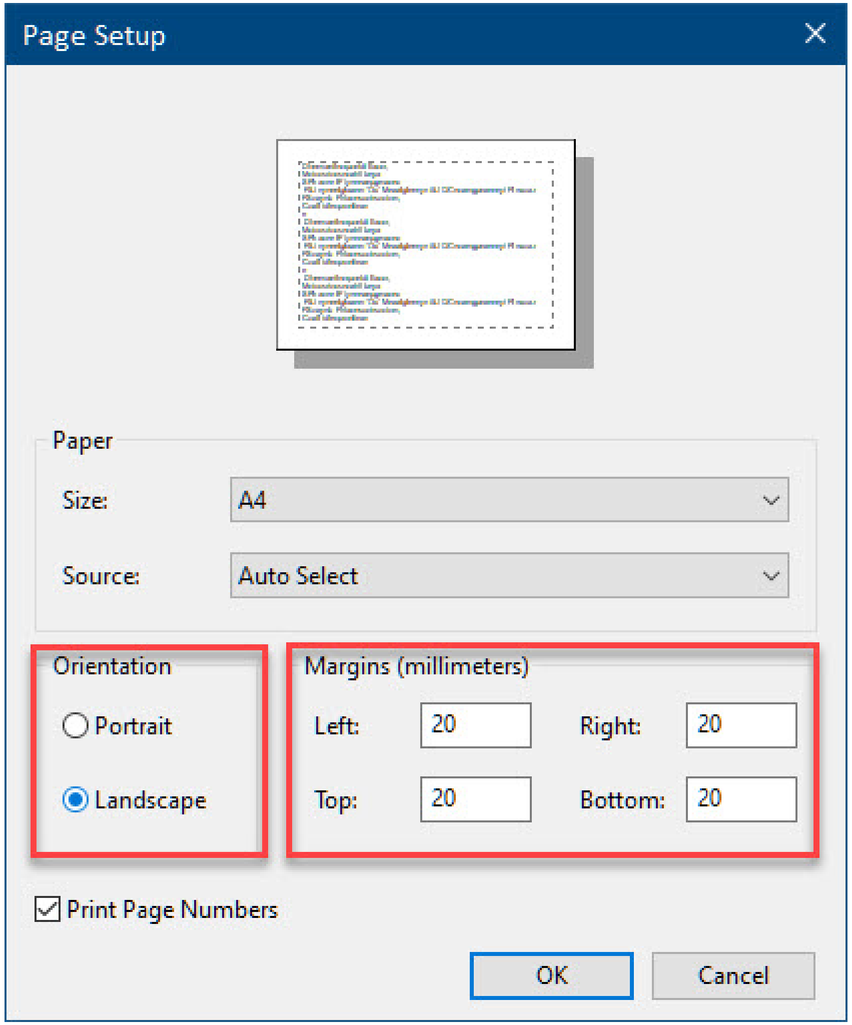
But, the output generated by the Fortran routine khb.exe is landscape, and fixed format width, with hard page-breaks (for printing purposes). So, you will need to quickly reformat the document in Wordpad - for readability and printing.

I use a Windows default monospaced font (Consolas) which is installed with Windows versions Vista and upwards. Font-size = 9.

So, **step 1** for the Wordpad document, **set the Page width to Landscape and set all margins to 20mm**



which shows:



Paper

Size: A4

Source: Auto Select

Orientation

○ Portrait

● Landscape

Margins (millimeters)

Left: 20 Right: 20

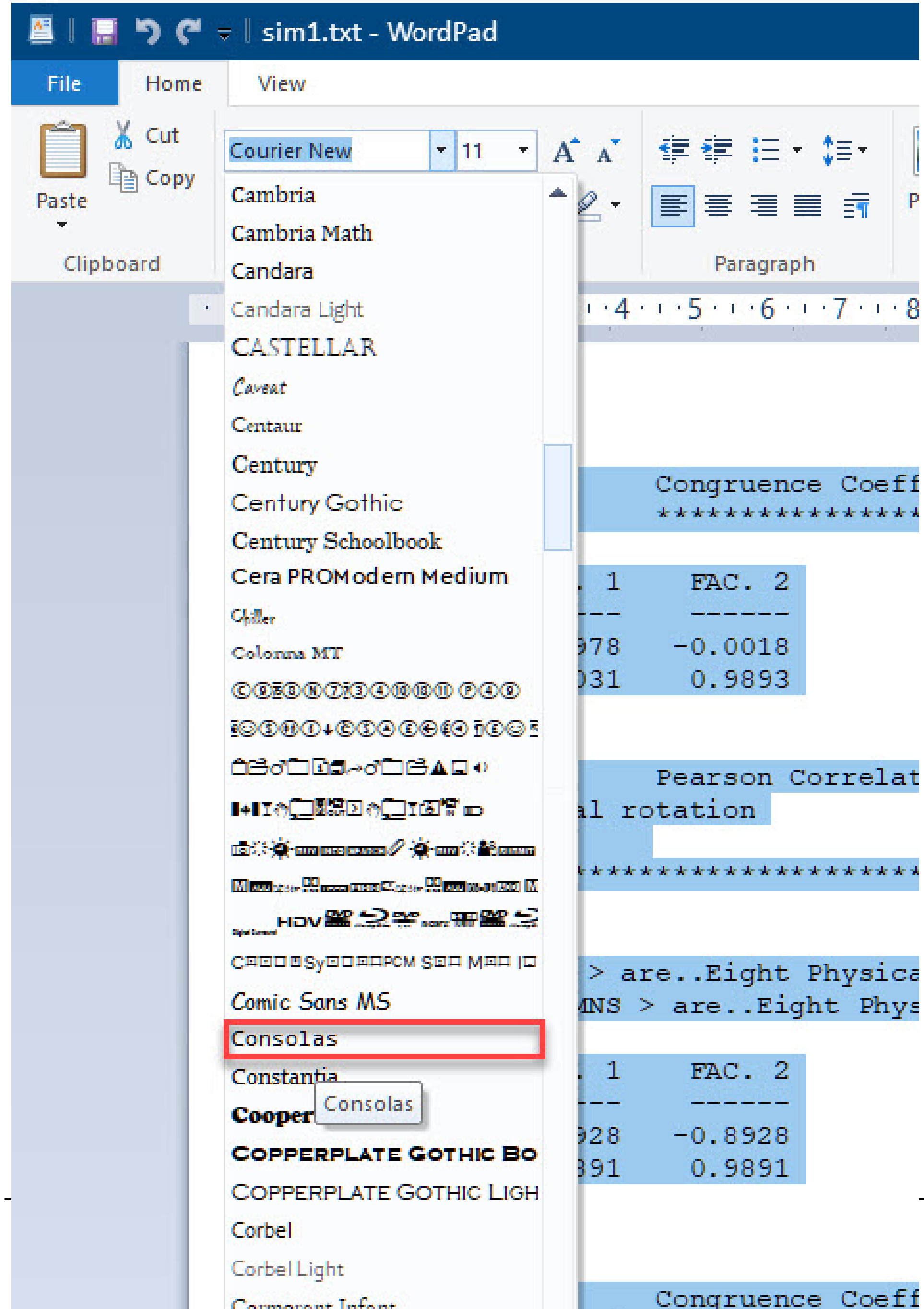
Top: 20 Bottom: 20

☑ Print Page Numbers

Then click OK ..

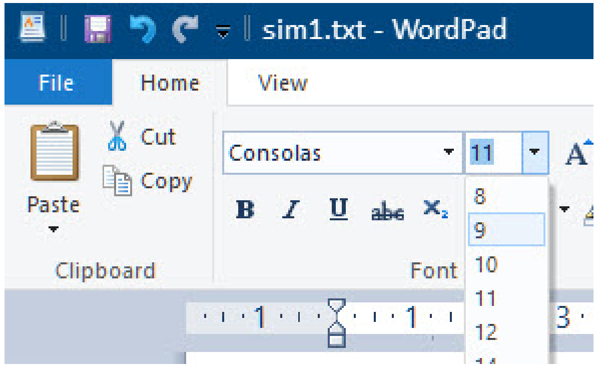
Step 2: change the font and font-size

Highlight all text in the document by pressing **Ctrl+A** (which highlights the entire document). Then select the font and size ..

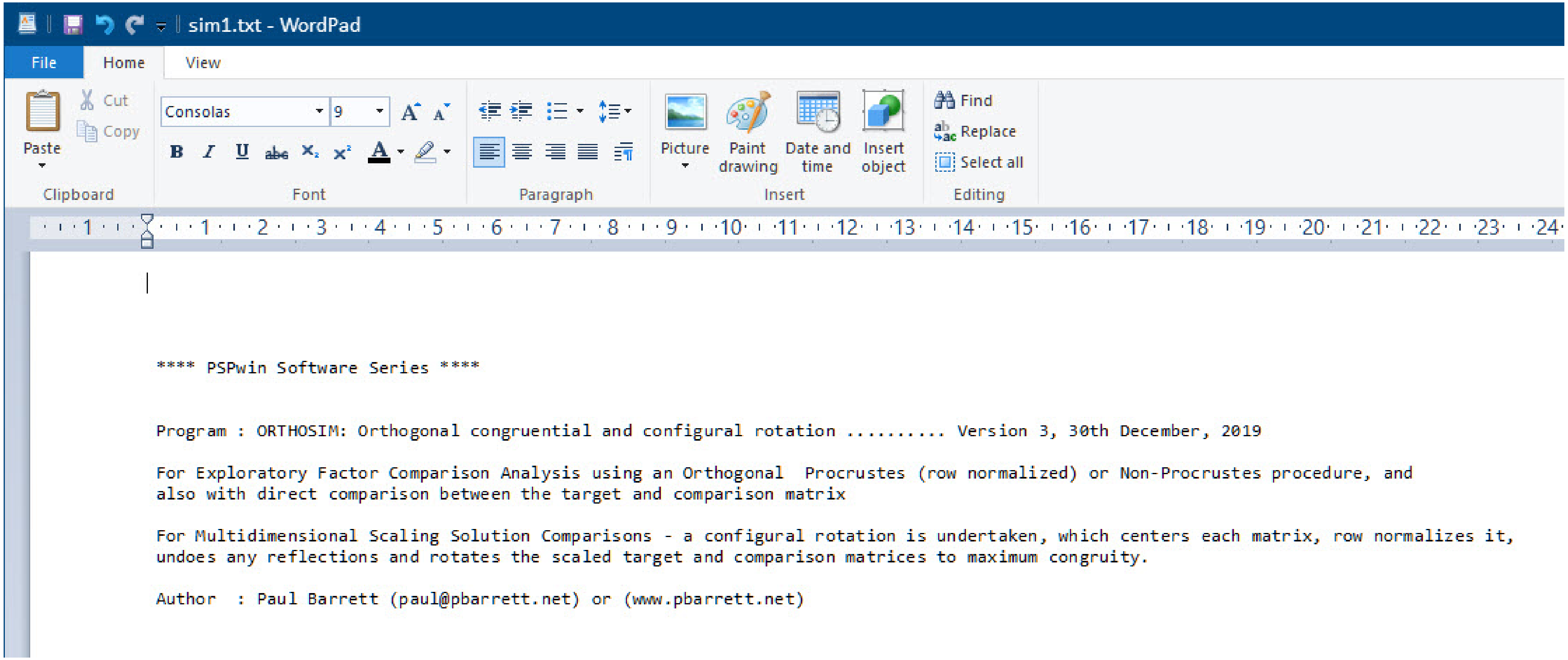


```
          Congruence Coeff
          ****************

. 1       FAC. 2
---       ------
978      -0.0018
031       0.9893


          Pearson Correlat
al rotation

**********************


 > are..Eight Physica
MNS > are..Eight Phys

. 1       FAC. 2
---       ------
928      -0.8928
891       0.9891


          Congruence Coeff
```

and size



which results in the document now looking like:



```
**** PSPwin Software Series ****


Program : ORTHOSIM: Orthogonal congruential and configural rotation .......... Version 3, 30th December, 2019

For Exploratory Factor Comparison Analysis using an Orthogonal  Procrustes (row normalized) or Non-Procrustes procedure, and
also with direct comparison between the target and comparison matrix

For Multidimensional Scaling Solution Comparisons - a configural rotation is undertaken, which centers each matrix, row normalizes it,
undoes any reflections and rotates the scaled target and comparison matrices to maximum congruity.

Author  : Paul Barrett (paul@pbarrett.net) or (www.pbarrett.net)
```

all titles and long-lines are now as 'designed'.

You can view and print the document from Wordpad, or copy areas etc to other documents as you like.

Again, sorry about the clumsy way I'm having to handle this, but this output was originally designed for 132 character, fan-fold-paper, dot-matrix printers! Nowadays I'd be using a rich-text output routine and automatic conversion to a pdf document 'on demand'. But this requires a complete reprogramming effort taking weeks, not days .. and I just don't have that kind of time I used to have as a university researcher/lecturer.

Anyway ..

If you wish to rerun the same analysis using say row normalization, select the appropriate analysis radio-button and click on the Go button again. Likewise, if you wish to retain the same target matrix but compare a new matrix to it - then just change the comparison file name by clicking on the "Comparison File?" button. Then press Go etc.

Also, if you wish to see the effect of changing the value of the smoothing coefficient on the KSD, just change its values with everything else as previously set (except you may wish to divert the output into a new output file) and press Go!

In short - this interface is built for usability and speed!

Paul Barrett, 30th December, 2019,   email: paul@pbarrett.net

Part

IV

4 Chapter 4: The Output

4.1 Understanding Orthosim Output

Brief Output

This is what is obtained when selecting an Orthosim Analysis with the Brief Input option active ...

The Header page to the output shows you which files have been input to the routine ...

```
**** PSPwin Software Series ****

Program : ORTHOSIM: Orthogonal congruential and configural rotation .......... Version 3, 30th December, 2019

For Exploratory Factor Comparison Analysis using an Orthogonal  Procrustes (row normalized) or Non-Procrustes procedure, and
also with direct comparison between the target and comparison matrix

For Multidimensional Scaling Solution Comparisons - a configural rotation is undertaken, which centers each matrix, row normalizes it,
undoes any reflections and rotates the scaled target and comparison matrices to maximum congruity.

Author  : Paul Barrett (paul@pbarrett.net) or (www.pbarrett.net)



Target file : C:\Facsim\Eight Physical Variables Matrix_2.vf

Datafile Title : Eight Physical Variables - Matrix No. 2


Comparison file : C:\Facsim\Eight Physical Variables Matrix_1.vf

Datafile Title : Eight Physical Variables - Matrix No. 1
```

```
 ***** Factor Analysis Unit-Vectors - No Procrustes (no row normalization)

 ***** Results format will be BRIEF
```

Then, the next two pages display the input Target and Comparison loading/coordinate matrices, and the respective factor/MDS dimension correlation matrix. The Target matrix is presented first, followed by the Comparison matrix. This is so that the user can be assured that the correct data are being read, prior to analysis.

```
                   Eight Physical Variables - Matrix No. 2
                   *******************************************************************************************


Input Format for Pattern Matrix is: (8F12.6)
Input Format for Factor Correlation Matrix is: (8F6.3)

                                        Input TARGET factor/coordinate matrix
                                        *************************************


             FAC. 1    FAC. 2
             ------    ------
Var_1       1.0000    0.0000
Var_2       1.0000    0.0000
Var_3       1.0000    0.0000
Var_4       1.0000    0.0000
Var_5       0.0000    1.0000
Var_6       0.0000    1.0000
Var_7       0.0000    1.0000
Var_8       0.0000    1.0000

 HYP.COUNT       4         4
 VARIANCE    4.0000    4.0000
```

*The HYP COUNT = the hyperplane count (the loadings/coordinate values <= |0.1|)

*   VARIANCE = the sum of squared loadings/coordinate values.

```
     Comparison Factor Intercorrelation Matrix:  Eight Physical Variables - Matrix No. 2

           FAC. 1    FAC. 2
           ------    ------
 FAC. 1    1.0000    0.0000
 FAC. 2    0.0000    1.0000

---------------------------------------------------------------------------------------------------
                    Eight Physical Variables - Matrix No. 1
                    *******************************************************************************************


Input Format for Pattern Matrix is: (8F12.6)
Input Format for Factor Correlation Matrix is: (8F6.3)

                                        Input COMPARISON factor/coordinate matrix
                                        *****************************************


             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8560   -0.3240
Var_2       0.8480   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.5100
Var_8       0.6070    0.3510

 HYP.COUNT       0         0
 VARIANCE    4.4481    1.5104



     Comparison Factor Intercorrelation Matrix:  Eight Physical Variables - Matrix No. 1

           FAC. 1    FAC. 2
           ------    ------
 FAC. 1    1.0000    0.0000
 FAC. 2    0.0000    1.0000
```

Then, the overall results page is presented. This details the overall solution congruence (which is the mean of the congruence coefficients calculated comparing each variable's values across all factors/coordinate axes). The "row" congruences are presented in a table below the summary results - I've rounded these row coefficients to two decimal places for simple convenience.Remember, the coefficient varies between -1.0 and +1.0.

Following this is the overall solution Double-Scaled Euclidean (DSED) similarity coefficient .. which includes the particular minimum and maximum value applied for these data. This coefficient varies between 0 and 1 with 1 = absolute identity. This is the mean of the individual factor/MDS coordinate dimension DSEDs.

Then comes the overall solution Kernel Distance Smoothed (KSD) similarity coefficient .. which includes the particular smoothing value used here (0.2). This is the mean of the individual factor/MDS coordinate dimension KSDs. Note how the small smoothing value here significantly reduces the apparent agreement. Generally, I use a smoothing value of about a fifth to a tenth of the minimum and maximum range ... so for -1 to +1 I'd maybe use a value of 0.2 up to 0.4 ... this needs calibrating by you in order to gauge how far you would like this coefficient to express the dissimilarity you see "by eye". Example 3 explains how I generally do this myself.

Note, both the DSED and KSD coefficients are sensitive to the scale discrepancy between the size of loadings in each matrix, unlike the congruence coefficient which is only sometimes sensitive (See the congruence coefficient help for examples of this phenomenon).

Ok - finally, we see the root-mean-square deviations, first between the input target and comparison matrix, then between the orthogonalized forms of each, then between the orthognalized target and maximally congruent comparison matrix. When presenting two orthogonal matrices to this routine, the orthogonalized forms are in fact the same matrices, hence no difference between the first two rms indices. You only see a difference when one or both matrices have been obliquely rotated. Then, orthogonalization has an effect - and is duly reported via the rms index. The 3rd rms index shows you how close the matrices have been brought into congruency - in terms of a root mean square deviation index. Obviously, the lower the rms value, the close are the two matrices to one another (in terms of least squares deviations of each variable across matrices).

```
              Matching Coefficients - by variable across dimensions (factors), and an overall mean congruence (of the variable
congruences)
              *************************************************************************************************************************
*

 ***** Factor Analysis Unit-Vectors - No Procrustes (no row normalization)
```

```
     Overall Solution Congruence = 0.96563

     Overall Solution Double-Scaled Euclidean Similarity =  0.89814          [Scaling Minimum =  -1.0000      Maximum =   1.0000 ]

     Overall Solution Kernel Distance Smoothed Similarity =  0.62774         [Scaling Smoother value =  0.20000 ]

     Root Mean Square deviation between the actual input target and original comparison matrix =  0.65427

     Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix =  0.65427

     Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix =  0.29105


          Congruence Coefficients calculated for each row (variable) in the target and maximally congruent comparison matrix
          ------------------------------------------------------------------------------------------------------------------

   variable     Name/ID          congruence    congruences rounded to two decimal places
   --------     -------          ----------    -----------------------------------------
        1       Var_1             0.95739        0.96
        2       Var_2             0.97956        0.98
        3       Var_3             0.98271        0.98
        4       Var_4             0.96526        0.97
        5       Var_5             0.96502        0.97
        6       Var_6             0.96798        0.97
        7       Var_7             0.98289        0.98
        8       Var_8             0.92425        0.92
```

If the congruences fall below certain values, then appropriate warnings will be given at the side of each variable:

If a congruence is less than or equal to **0.4**, then the message "**Trivial Variable Similarity**" is printed alongside the variable congruence.
If a congruence is less than or equal to **0.6** but greater than **0.4**, then the message "**Bad Match**" is printed alongside the variable congruence.
If a congruence is less than or equal to **0.8** but greater than **0.6** then the message "**Poor Match**" is printed alongside the variable congruence.

What comes next are the summary values for several "agreement" indices ... The first table shows the DSED and KSD coefficients for each factor/MDS coordinate dimension.

```
          Double-Scaled (DSES) and Kernel Smoothed Euclidean Similarity Coefficients - target vs comparison factors/coordinate axes
          *************************************************************************************************************************

         The coefficients range from 0 = maximum dissimilarity to 1 = absolute identity

    Factor #     DSES Coefficient     Kernel Smoothed Similarity
    --------     ----------------     --------------------------
       1             0.91273                0.70956
       2             0.88355                0.54591
```

The next one shows the Pearson correlations between each factor/MDS coordinate dimension.

```
                  Pearson Correlations:  between the target and maximally congruent comparison matrix
                  ************************************************************************************

   The < ROWS > are..Eight Physical Variables - Matrix No. 2
   The < COLUMNS > are..Eight Physical Variables - Matrix No. 1

          FAC. 1    FAC. 2
          ------    ------
FAC. 1    0.9938   -0.9689
FAC. 2   -0.9938    0.9689
```

Then we see the congruence coefficients between each factor/MDS coordinate dimension.

```
                  Congruence Coefficients:  between the target and maximally congruent comparison matrix
                  **************************************************************************************

          FAC. 1    FAC. 2
          ------    ------
FAC. 1    0.9706    0.2655
FAC. 2    0.2337    0.9560
```

Next we see two tables - showing Pearson and congruence coefficients - between the initial input target and comparison matrices, These tables show you what the agreement was like between the matrices, **prior to the congruential rotation procedure**. Sometimes interesting to see -

especially when looking at the effect of simple structure rotation and/or oblique rotations on the matrices, compared to the orthogonal congruity found using the least-squares discrepancy criterion.

```
                  Pearson Correlations: between the original input target and comparison matrices - prior to any congruential rotation
                  ********************************************************************************************************************

   The < ROWS > are..Eight Physical Variables - Matrix No. 2
   The < COLUMNS > are..Eight Physical Variables - Matrix No. 1

          FAC. 1    FAC. 2
          ------    ------
FAC. 1    0.8928   -0.9891
FAC. 2   -0.8928    0.9891



                  Congruence Coefficients: between the original input target and comparison matrices - prior to any congruential rotation
                  ***********************************************************************************************************************

          FAC. 1    FAC. 2
          ------    ------
FAC. 1    0.7925   -0.6050
FAC. 2    0.6062    0.7828


        --- Finished ---
```

Long Output

This is what is obtained when selecting an Orthosim Analysis with the Long Input option active ...

After the first two pages of input matrix output as above, the next page/s are displayed prior to the Overall Solution index results ... These are the actual "results" matrices - the target and the maximally congruent comparison matrix over which all subsequent summary indices are computed. Here you can see just how close (if at all) the comparison matrix loadings/values look compared to the target matrix loadings (note, this example uses a synthetic target matrix).

```
   Orthogonalised Input Target Matrix: Eight Physical Variables - Matrix No. 2


             FAC. 1    FAC. 2
             ------    ------
Var_1       1.0000    0.0000
Var_2       1.0000    0.0000
Var_3       1.0000    0.0000
Var_4       1.0000    0.0000
Var_5       0.0000    1.0000
Var_6       0.0000    1.0000
Var_7       0.0000    1.0000
Var_8       0.0000    1.0000

 HYP.COUNT       4         4
 VARIANCE    4.0000    4.0000
```

```
   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 1

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8763    0.2643
Var_2       0.9235    0.1896
Var_3       0.8900    0.1677
Var_4       0.8674    0.2348
Var_5       0.2471    0.9096
Var_6       0.2009    0.7745
Var_7       0.1407    0.7510
Var_8       0.2677    0.6481

 HYP.COUNT       0         0
 VARIANCE    3.3580    2.6005
```

Then all other output is the same as for the brief option .. so all that really differs between the options is whether you get to see the orthogonalised matrices used as a target and comparison.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

Part V

5 Chapter 5: Orthosim, Scaling, and Order

5.1 Example 1: Perfect monotonicity but size discrepancy

This example shows the effects of "procrustes" row normalisation (stretching or shrinking values in a matrix so as to express both matrices in the same metric). - or raw matrix comparisons (without row normalization)

We'll start off gently at first! Here I use two matrices ... (these are installed for you in **C:\Users\Public\Orthosim**)

Target Datafile : C:\Facsim\Eight Physical Variables Matrix_2 - adjusted.vf

Datafile Title : Eight Physical Variables - Matrix No. 2 - adjusted loadings

Comparison Datafile : C:\Facsim\Eight Physical Variables Matrix_1.vf

Datafile Title : Eight Physical Variables - Matrix No. 1

The input matrices are:

```
Input TARGET factor/coordinate matrix
*************************************
             FAC. 1    FAC. 2
             ------    ------
Var_1       0.1000    0.1000
Var_2       0.2000    0.1000
Var_3       0.3000    0.1000
Var_4       0.4000    0.1000
Var_5       0.1000    0.1000
Var_6       0.1000    0.2000
Var_7       0.1000    0.2000
Var_8       0.0000    0.9000

Input COMPARISON factor/coordinate matrix
*****************************************
             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8560   -0.3240
Var_2       0.8480   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.5100
Var_8       0.6070    0.3510
```

The Orthosim (non-procrustes) target and maximally congruent comparison matrix are:

```
   Orthogonalised Input Target Matrix: Eight Physical Variables - Matrix No. 2 - adjusted loadings

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.1000    0.1000
Var_2       0.2000    0.1000
Var_3       0.3000    0.1000
Var_4       0.4000    0.1000
Var_5       0.1000    0.1000
Var_6       0.1000    0.2000
Var_7       0.1000    0.2000
Var_8       0.0000    0.9000


   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 1

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8392    0.3653
Var_2       0.8949    0.2967
Var_3       0.8641    0.2710
Var_4       0.8338    0.3350
Var_5       0.1387    0.9324
Var_6       0.1086    0.7927
Var_7       0.0516    0.7624
Var_8       0.1898    0.6750
```

Overall Solution congruence = **0.94329**

The factor by factor congruence coefficients computed from the above two matrices are:

```
           FAC. 1    FAC. 2
           ------    ------
 FAC. 1    0.8875    0.5683
 FAC. 2    0.3325    0.6847
```

Now look at the corresponding output form the routine (where I have exposed the actual "procrustean" matrices which are compared to one another, and from which the congruence coefficients are calculated.

```
   ROW NORMALISED (original factor space) ORTHOGONAL MULTIPLE GROUP MATRIX #1
   ==========================================================================
             FAC. 1    FAC. 2
             ------    ------
Var_1       0.7071    0.7071
Var_2       0.8944    0.4472
Var_3       0.9487    0.3162
Var_4       0.9701    0.2425
Var_5       0.7071    0.7071
Var_6       0.4472    0.8944
Var_7       0.4472    0.8944
Var_8       0.0000    1.0000

   ROW NORMALISED (original factor space) ORTHOGONAL MULTIPLE GROUP MATRIX #2
   ==========================================================================
             FAC. 1    FAC. 2
             ------    ------
Var_1       0.9474    0.3200
Var_2       0.9724    0.2332
Var_3       0.9761    0.2173
Var_4       0.9562    0.2928
Var_5       0.2304    0.9731
Var_6       0.2192    0.9757
Var_7       0.1518    0.9884
Var_8       0.3513    0.9363
```

Overall Solution congruence = **0.94669**

The factor by factor congruence coefficients computed from the above two matrices are:

```
           FAC. 1    FAC. 2
           ------    ------
 FAC. 1    0.9307    0.6164
 FAC. 2    0.6294    0.9628
```

These factor by factor coefficients are a lot different in this example to those computed from a non-procrustean orthogonal target rotation.

BUT - it gets worse! What if I compare the original 8 physical variable matrix to one in which all loadings are divided by 100?
The input file is: Eight Physical Variables Matrix divided by 100.vf (this file is installed for you in **C:\Users\Public\Orthosim**)

```
 ***** Factor Analysis Unit-Vectors - No Procrustes (no row normalization)

 ***** Results format will be LONG and DETAILED!

                    Eight Physical Variables - Matrix No. 1
                    *******************************************************************************************

Input Format for Pattern Matrix is: (8F12.6)
Input Format for Factor Correlation Matrix is: (8F6.3)

                                        Input TARGET factor/coordinate matrix
                                        *************************************


             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8560   -0.3240
Var_2       0.8480   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.5100
```

```
Var_8       0.6070    0.3510

 HYP.COUNT       0         0
 VARIANCE    4.4481    1.5104


     Target Factor Intercorrelation Matrix:  Eight Physical Variables - Matrix No. 1

           FAC. 1    FAC. 2
           ------    ------
 FAC. 1    1.0000    0.0000
 FAC. 2    0.0000    1.0000


                    Eight Physical Variables - Matrix No. 1 - all loadings divided by 100
                    *******************************************************************************************

Input Format for Pattern Matrix is: (8F14.6)
Input Format for Factor Correlation Matrix is: (8F6.3)

                                        Input COMPARISON factor/coordinate matrix
                                        *****************************************


             FAC. 1    FAC. 2
             ------    ------
Var_1       0.0086   -0.0032
Var_2       0.0085   -0.0041
Var_3       0.0081   -0.0041
Var_4       0.0083   -0.0034
Var_5       0.0075    0.0057
Var_6       0.0063    0.0049
Var_7       0.0057    0.0051
Var_8       0.0061    0.0035

 HYP.COUNT       8         8
 VARIANCE    0.0004    0.0002
```

```
     Comparison Factor Intercorrelation Matrix:  Eight Physical Variables - Matrix No. 1 - all loadings divided by 100

           FAC. 1    FAC. 2
           ------    ------
 FAC. 1    1.0000    0.0000
 FAC. 2    0.0000    1.0000
```

```
   Orthogonalised Non-Procrustes EFA Input Target Matrix: Eight Physical Variables - Matrix No. 1

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8560   -0.3240
Var_2       0.8480   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.5100
Var_8       0.6070    0.3510

 HYP.COUNT       0         0
 VARIANCE    4.4481    1.5104
```

```
   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 1 - all loadings divided by 100

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.0086   -0.0032
Var_2       0.0085   -0.0041
Var_3       0.0081   -0.0041
Var_4       0.0083   -0.0034
Var_5       0.0075    0.0057
Var_6       0.0063    0.0049
Var_7       0.0057    0.0051
Var_8       0.0061    0.0035

 HYP.COUNT       8         8
```

```
VARIANCE    0.0004    0.0002
```

```
              Matching Coefficients - by variable across dimensions (factors), and an overall mean congruence (of the variable
congruences)
              *****************************************************************************************************************
*

 ***** Factor Analysis Unit-Vectors - No Procrustes (no row normalization)

     Overall Solution Congruence = 1.00000

     Overall Solution Double-Scaled Euclidean Similarity =  0.70791          [Scaling Minimum =  -1.0000       Maximum =   1.0000 ]

     Overall Solution Kernel Distance Smoothed Similarity =  0.07128         [Scaling Smoother value =  0.20000 ]

     Root Mean Square deviation between the actual input target and original comparison matrix =  0.85440

     Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix =  0.85440

     Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix =  0.85440


          Congruence Coefficients calculated for each row (variable) in the target and maximally congruent comparison matrix
          -------------------------------------------------------------------------------------------------------------------

   variable     Name/ID          congruence     congruences rounded to two decimal places
   --------     -------          ----------     ----------------------------------------
        1       Var_1             1.00000         1.00
        2       Var_2             1.00000         1.00
        3       Var_3             1.00000         1.00
        4       Var_4             1.00000         1.00
        5       Var_5             1.00000         1.00
        6       Var_6             1.00000         1.00
        7       Var_7             1.00000         1.00
        8       Var_8             1.00000         1.00
```

```
          Double-Scaled (DSES) and Kernel Smoothed Euclidean Similarity Coefficients - target vs comparison factors/coordinate axes
          *************************************************************************************************************************

         The coefficients range from 0 = maximum dissimilarity to 1 = absolute identity

    Factor #     DSES Coefficient     Kernel Smoothed Similarity
    --------     ----------------     --------------------------
       1             0.63090                0.00492
       2             0.78492                0.13763


                  Pearson Correlations:  between the target and maximally congruent comparison matrix
                  ************************************************************************************

   The < ROWS > are..Eight Physical Variables - Matrix No. 1
   The < COLUMNS > are..Eight Physical Variables - Matrix No. 1 - all loadings divided by 100

          FAC. 1    FAC. 2
          ------    ------
FAC. 1    1.0000   -0.8478
FAC. 2   -0.8478    1.0000

                  Congruence Coefficients:  between the target and maximally congruent comparison matrix
                  **************************************************************************************

          FAC. 1    FAC. 2
          ------    ------
FAC. 1    1.0000    0.0002
FAC. 2    0.0002    1.0000
```

This example shows why I recommend only using a congruence coefficient as a matching index alongside a DSED or KSD coefficient. Even where one matrix contains essentially all zero loadings - the mere fact that a perfect monotonic order is maintained in the 3rd decimal place is sufficient to produce congruence and pearson coefficients of 1.0.

Now, let's run the same analysis using a procrustes row normalization ...

first we see the row-normalized version of our input target matrix.

```
   Orthogonalised EFA Procrustes (Row Normalized Version) Input Target Matrix

   Eight Physical Variables - Matrix No. 1

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.9352   -0.3540
Var_2       0.8995   -0.4370
Var_3       0.8922   -0.4516
Var_4       0.9247   -0.3806
Var_5       0.7957    0.6058
Var_6       0.7886    0.6149
Var_7       0.7447    0.6674
Var_8       0.8657    0.5006

 HYP.COUNT       0         0
 VARIANCE    5.8938    2.1062
```

Now, the matrix which only had loadings defined in the 3rd decimal place is now itself row normalized and rotated into maximum congruity with the target row-normalized matrix ...

```
   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 1 - all loadings divided by 100

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.9352   -0.3540
Var_2       0.8995   -0.4370
Var_3       0.8922   -0.4516
Var_4       0.9247   -0.3806
Var_5       0.7957    0.6058
Var_6       0.7886    0.6149
```

```
Var_7       0.7447    0.6674
Var_8       0.8657    0.5006

 HYP.COUNT       0         0
 VARIANCE    5.8938    2.1062


              Matching Coefficients - by variable across dimensions (factors), and an overall mean congruence (of the variable
congruences)
              ****************************************************************************************************************************
*

 ***** Factor Analysis Unit-Vectors - Procrustes Fit

     Overall Solution Congruence = 1.00000

     Overall Solution Double-Scaled Euclidean Similarity =  1.00000          [Scaling Minimum =  -1.0000      Maximum =   1.0000 ]

     Overall Solution Kernel Distance Smoothed Similarity =  1.00000         [Scaling Smoother value =  0.20000 ]

     Root Mean Square deviation between the actual input target and original comparison matrix =  0.85440

     Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix =  0.00000

     Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix =  0.00000


          Congruence Coefficients calculated for each row (variable) in the target and maximally congruent comparison matrix
          ----------------------------------------------------------------------------------------------------------------------

   variable     Name/ID          congruence     congruences rounded to two decimal places
   --------     -------          ----------     ----------------------------------------
        1       Var_1             1.00000         1.00
        2       Var_2             1.00000         1.00
        3       Var_3             1.00000         1.00
        4       Var_4             1.00000         1.00
        5       Var_5             1.00000         1.00
        6       Var_6             1.00000         1.00
        7       Var_7             1.00000         1.00
        8       Var_8             1.00000         1.00
```

```
      Double-Scaled (DSES) and Kernel Smoothed Euclidean Similarity Coefficients - target vs comparison factors/coordinate axes
      *************************************************************************************************************************

     The coefficients range from 0 = maximum dissimilarity to 1 = absolute identity

Factor #     DSES Coefficient     Kernel Smoothed Similarity
--------     ----------------     --------------------------
   1             1.00000                1.00000
   2             1.00000                1.00000
```

As can be seen, once the matrices are row-normalized into the same metric, then they are exactly the same as one-another. The question is, what is required to be compared, the original matrices "as is" or their unified metric versions? This is not so easy to answer. Ordinarily, psychologists/researchers interpret factor matrix loadings of the original matrices - not row-normalized ones. Yet, a simple transformation of the matrices yields a completely different set of loadings and absolute identity between matrices. My feeling is that **IF** the original factor matrices are to be compared "as is" - then size does indeed matter and you cannot simply stretch or shrink a comparison matrix loadings to another. With MDS coordinates, the issue is not relevant - as these coordinates are essentially scale free. True configural similarity is required - rather than factor similarity. MDS solutions are arbitrary in terms of their location, scale, and orientation of variables in geometric space.It is the distance relations between variables which are critical in MDS; such relations can be preserved while allowing the origin, scale, and reflection of solutions to vary. Hence, the extra transformations required prior to congruential rotation in MDS.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

5.2 Example 2: Which is the target? Order matters

Ah yes - the old "order effect". **Lost in the small print in many descriptions of congruential rotation is that the order in which you input matrices matters.** That is, which matrix you define as the target matrix matters to the factor/coordinate axis vs factor/coordinate axis comparisons. For example, let's run the 8 physical variable example ... The order of matrices is:

Target Datafile : C:\Facsim\Eight Physical Variables Matrix**_2**.vf
Datafile Title : Eight Physical Variables - Matrix No. 2

Comparison Datafile : C:\Facsim\Eight Physical Variables Matrix**_1**.vf

Datafile Title : Eight Physical Variables - Matrix No. 1

We obtain the following overall solution indices ...

Overall Solution Congruence = **0.96563**

Overall Solution Double-Scaled Euclidean Similarity = **0.89814** [Scaling Minimum = -1.0000 Maximum = 1.0000]

Overall Solution Kernel Distance Smoothed Similarity = **0.62774** [Scaling Smoother value = 0.20000]

Root Mean Square deviation between the actual input target and original comparison matrix = 0.65427

Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix = 0.65427

Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix = 0.29105

with factor by factor results of:

```
       Congruence Coefficients calculated for each row (variable) in the target and maximally congruent comparison matrix
       ------------------------------------------------------------------------------------------------------------------

variable     Name/ID          congruence     congruences rounded to two decimal places
--------     -------          ----------     ----------------------------------------
```

```
       1       Var_1             0.95739         0.96
       2       Var_2             0.97956         0.98
       3       Var_3             0.98271         0.98
       4       Var_4             0.96526         0.97
       5       Var_5             0.96502         0.97
       6       Var_6             0.96798         0.97
       7       Var_7             0.98289         0.98
       8       Var_8             0.92425         0.92


                  Double-Scaled Euclidean Similarity (DSES) Coefficients - target vs comparison factors/coordinate axes
                  ****************************************************************************************************

         The coefficients range from 0 = maximum dissimilarity to 1 = absolute identity

    Factor #     DSES Coefficient     Kernel Smoothed Similarity
    --------     ----------------     --------------------------
       1             0.91273                0.70956
       2             0.88355                0.54591


                  Congruence Coefficients:  between the target and maximally congruent comparison matrix
                  **************************************************************************************

          FAC. 1    FAC. 2
          ------    ------
FAC. 1    0.9706    0.2655
FAC. 2    0.2337    0.9560
```

Now we input the matrices with Matrix No. 1 as our target, and Matrix No. 2 as our comparison matrix ...

Target Datafile : C:\Facsim\Eight Physical Variables Matrix**_1.**vf
Datafile Title : Eight Physical Variables - Matrix No. 1

Comparison Datafile : C:\Facsim\Eight Physical Variables Matrix_**2**.vf

Datafile Title : Eight Physical Variables - Matrix No. 2

We obtain the following overall solution indices ...

Overall Solution Congruence = **0.96563**

Overall Solution Double-Scaled Euclidean Similarity = **0.91344** [Scaling Minimum = -1.0000 Maximum = 1.0000]

Overall Solution Kernel Distance Smoothed Similarity = **0.68490** [Scaling Smoother value = 0.20000]

Root Mean Square deviation between the actual input target and original comparison matrix = 0.65427

Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix = 0.65427

Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix = 0.29105

Note the marginal change in the DSED and KSD coefficients - reflecting the sensitivity of these coefficients to the changed target and maximally congruent factor pattens, but no change between the Congruence coefficient or root-mean-square coefficients. These latter coefficients (the rms) should not change as they simply index the least-squares fit - which will be the same regardless of which matrix is input first.
However, look at the factor by factor results ...

```
          Congruence Coefficients calculated for each row (variable) in the target and maximally congruent comparison matrix
          ---------------------------------------------------------------------------------------------------------------

   variable     Name/ID          congruence     congruences rounded to two decimal places
   --------     -------          ----------     ----------------------------------------
        1       Var_1             0.95739         0.96
        2       Var_2             0.97956         0.98
        3       Var_3             0.98271         0.98
        4       Var_4             0.96526         0.97
        5       Var_5             0.96502         0.97
        6       Var_6             0.96798         0.97
        7       Var_7             0.98289         0.98
        8       Var_8             0.92425         0.92
```

No change in the row-by-row congruences ..but changes to the distance and factor-by-factor congruences ...

```
                   Double-Scaled Euclidean Similarity (DSES) Coefficients - target vs comparison factors/coordinate axes
                   ****************************************************************************************************

          The coefficients range from 0 = maximimum dissimilarity to 1 = absolute identity

     Factor #     DSES Coefficient     Kernel Smoothed Similarity
     --------     ----------------     --------------------------
        1             0.96909                0.95607
        2             0.85780                0.41373


                   Congruence Coefficients:  between the target and maximally congruent comparison matrix
                   **************************************************************************************

           FAC. 1    FAC. 2
           ------    ------
 FAC. 1    0.9978   -0.0018
 FAC. 2   -0.0031    0.9893
```

So- order of entry (which is the target and which is the comparison matrix) **WILL make a difference** to the size of the factor-by-factor congruencecoefficients, as well as to the DSED and KSD coefficients.It will not make a difference to the overall mean solution congruence.

The overall similarity DSED and KSD indices are relatively unaffected by the order of entry of the matrices - but the factor-by-factor indices are sensitive, as are the congruence coefficients.

So - be careful!

Paul Barrett, 30th December, 2019,   email: paul@pbarrett.net

5.3 Example 3: Kernel Distance smoothing - calibration #1

Ok - how to scale the coefficient ... well, run an test analysis such as in example 1 ...

Target file : C:\Facsim\Eight Physical Variables Matrix_1.vf
Datafile Title : Eight Physical Variables - Matrix No. 1

Comparison file : C:\Facsim\Eight Physical Variables Matrix_001.vf
Datafile Title : Eight Physical Variables - Matrix No. 1 four loadings set to 0.001

The file "Eight Physical Variables Matrix_001.vf" has exactly the same loadings as the target file - except four have been set to 0.001. I run a non-procrustes rotation ...

I'm using **a KSD value of 0.2** ...

The input matrices are:

```
                    Eight Physical Variables - Matrix No. 1
                    *******************************************************************************************

                                        Input TARGET factor/coordinate matrix
                                        *************************************


             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8560   -0.3240
Var_2       0.8480   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.5100
Var_8       0.6070    0.3510

 HYP.COUNT       0         0
 VARIANCE    4.4481    1.5104
```

```
                    Eight Physical Variables - Matrix No. 1 four loadings set to 0.001-
                    ******************************************************************************************

                                        Input COMPARISON factor/coordinate matrix
                                        *****************************************

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.0010   -0.3240
Var_2       0.0010   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.0010
Var_8       0.6070    0.0010

 HYP.COUNT       2         2
 VARIANCE    2.9963    1.1271


   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 1 four loadings set to 0.001-

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.0864   -0.3123
Var_2       0.1096   -0.3972
Var_3       0.8873   -0.1814
Var_4       0.8918   -0.1107
Var_5       0.5729    0.7486
Var_6       0.4789    0.6410
Var_7       0.5486    0.1510
Var_8       0.5852    0.1610
```

The overall results are:

Matching Coefficients - by variable across dimensions (factors), and an overall mean congruence (of the variable congruences)
****
****

***** Factor Analysis Unit-Vectors - No Procrustes (no row normalization)

Overall Solution Congruence = **0.87152**

Overall Solution Double-Scaled Euclidean Similarity = 0.85273 [Scaling Minimum = -1.0000 Maximum = 1.0000]

Overall Solution Kernel Distance Smoothed Similarity = **0.66215** [Scaling Smoother value = **0.20000**]

Root Mean Square deviation between the actual input target and original comparison matrix = 0.47828

Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix = 0.47828

Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix = 0.43691

```
       Congruence Coefficients calculated for each row (variable) in the target and maximally congruent comparison matrix
       ------------------------------------------------------------------------------------------------------------------

variable     Name/ID          congruence     congruences rounded to two decimal places
--------     -------          ----------     ----------------------------------------
     1       Var_1             0.59061         0.59               Bad Match
     2       Var_2             0.66057         0.66               Poor Match
     3       Var_3             0.96460         0.96
     4       Var_4             0.96460         0.96
     5       Var_5             0.96460         0.96
     6       Var_6             0.96460         0.96
     7       Var_7             0.89510         0.90
     8       Var_8             0.96748         0.97
```

Determine whether or not you feel the "distance degredation" of the KSD coefficient is too harsh or too "soft" - if you feel it's a bit too harsh in terms of its value here, try re-running the same analysis with the smoothing parameter set at a higher value - say 0.5 ... the result of this is:

Overall Solution Kernel Distance Smoothed Similarity = **0.86952** [Scaling Smoother value = **0.50000**]

Perhaps this feels about right? Alternatively, you may feel that's it's still a bit generous ....so you set it to **0.35** ...

Overall Solution Kernel Distance Smoothed Similarity = **0.80009** [Scaling Smoother value = **0.35000**]

A final adjustment

It's probably as well to check how this performs on another adjusted matrix where the loadings are not that similar but where the monotonicity is maintained to a degree ...

```
Target file : C:\Facsim\Eight Physical Variables Matrix_1.vf
Datafile Title : Eight Physical Variables - Matrix No. 1

Comparison file : C:\Facsim\Eight Physical Variables Matrix_2 - adjusted.vf

Datafile Title : Eight Physical Variables - Matrix No. 2 - adjusted loadings
```

```
                                        Input TARGET factor/coordinate matrix
                                        *************************************


             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8560   -0.3240
Var_2       0.8480   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.5100
Var_8       0.6070    0.3510
```

```
 HYP.COUNT       0         0
 VARIANCE    4.4481    1.5104


                    Eight Physical Variables - Matrix No. 2 - adjusted loadings
                    *******************************************************************************************

                                        Input COMPARISON factor/coordinate matrix
                                        *****************************************

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.1000    0.1000
Var_2       0.2000    0.1000
Var_3       0.3000    0.1000
Var_4       0.4000    0.1000
Var_5       0.1000    0.1000
Var_6       0.1000    0.2000
Var_7       0.1000    0.2000
Var_8       0.0000    0.9000

 HYP.COUNT       5         5
 VARIANCE    0.3300    0.9400


   Orthogonalised Non-Procrustes EFA Input Target Matrix: Eight Physical Variables - Matrix No. 1


             FAC. 1    FAC. 2
             ------    ------
Var_1       0.8560   -0.3240
Var_2       0.8480   -0.4120
Var_3       0.8080   -0.4090
Var_4       0.8310   -0.3420
Var_5       0.7500    0.5710
Var_6       0.6310    0.4920
Var_7       0.5690    0.5100
Var_8       0.6070    0.3510
```

```
 HYP.COUNT       0         0
 VARIANCE    4.4481    1.5104



   Orthogonal Maximally Congruent Comparison Matrix: Eight Physical Variables - Matrix No. 2 - adjusted loadings

             FAC. 1    FAC. 2
             ------    ------
Var_1       0.1414    0.0018
Var_2       0.2130   -0.0680
Var_3       0.2847   -0.1377
Var_4       0.3563   -0.2075
Var_5       0.1414    0.0018
Var_6       0.2112    0.0735
Var_7       0.2112    0.0735
Var_8       0.6281    0.6446

 HYP.COUNT       0         5
 VARIANCE    0.7770    0.4930


             Matching Coefficients - by variable across dimensions (factors), and an overall mean congruence (of the variable congruences)
             *****************************************************************************************************************************

 ***** Factor Analysis Unit-Vectors - No Procrustes (no row normalization)

     Overall Solution Congruence = 0.94329

     Overall Solution Double-Scaled Euclidean Similarity =  0.77991          [Scaling Minimum =  -1.0000       Maximum =   1.0000 ]

     Overall Solution Kernel Distance Smoothed Similarity =  0.51217         [Scaling Smoother value =  0.35000 ]

     Root Mean Square deviation between the actual input target and original comparison matrix =  0.73562

     Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix =  0.73562
     Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix =  0.63050
```

Maybe the DSED suits you best? The overall solution congruence coefficient certainly looks a little high at 0.94 given the obvious magnitude discrepancies between the target and orthogonally congruent matrices.

Anyway, I hope the above helped clarify somewhat the effect of the smoothing coefficient.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

5.4 Example 4: Using a synthetic orthogonal target matrix

Well, the input target matrix for 8 physical variables example is relevant here …

The file itself is: Eight Physical Variables Matrix_2.vf

```
Eight Physical Variables - Matrix No. 2
 2  8
Var_1   Var_2   Var_3   Var_4   Var_5   Var_6   Var_7   Var_8
(8F12.6)
(8F6.3)
       1.0         1.0         1.0         1.0         1.0         0.0        0.0       0.0
       0.0         0.0         0.0         0.0         1.0         1.0        1.0       1.0
  1.000  0.000
  0.000  1.000
```

which displays as:

```
             FAC. 1    FAC. 2
             ------    ------
Var_1       1.0000    0.0000
Var_2       1.0000    0.0000
Var_3       1.0000    0.0000
Var_4       1.0000    0.0000
Var_5       0.0000    1.0000
```

```
Var_6       0.0000    1.0000
Var_7       0.0000    1.0000
Var_8       0.0000    1.0000
```

This kind of synthentic target is useful when trying to fit the "scorekey" of a questionnaire say, to a set of factor loadings, or even an "ideal" factor pattern or idealised data model drawn from theory. You could use my small utility program "Target Matrix Generator" to easily create such files.

The example output in the Understanding Orthosim Output section shows the results of such synthetic target fitting.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

5.5 Example 5: Inputting obliquely rotated matrices

This example shows you the output when two obliquely rotated matrices are input to Orthosim ... (these are installed for you in **C:\Users\Public\Orthosim**)

Target Datafile : C:\Facsim\epqr48m.vf
Datafile Title : EPQR Short Form - item numbers referred to 100 item EPQR - MALES ONLY

Comparison Datafile : C:\Facsim\epqr48f.vf
Datafile Title : EPQR Short Form - item numbers referred to 100 item EPQR - FEMALES only

The short-form EPQR data for the males serves at the target, with the females being compared to these ...

The Input target matrix which has previously been obliquely rotated via direct oblimin looks like ... (a section of)

```
             FAC. 1    FAC. 2    FAC. 3    FAC. 4
             ------    ------    ------    ------
 Psy   5     0.1177   -0.2026    0.0982   -0.4767
 Psy   7    -0.1476   -0.1991   -0.1012   -0.3404
 Psy  25    -0.1508    0.1537   -0.1554   -0.3272
 Psy  29    -0.0064    0.0438   -0.0853   -0.4474
 Psy  41    0.1842    0.0154   -0.0733   -0.4402
 Psy  48    0.0336    0.0278    0.1543   -0.3619
 Psy  54    0.3581    0.0282    0.0344   -0.4601
 Psy  59   -0.1251   -0.0965   -0.0356   -0.4145
 Psy  75   -0.0411    0.0374   -0.0290   -0.3730
 Psy  79   -0.0635    0.0205   -0.0685   -0.4615
 Psy  88   -0.0139    0.0328   -0.1399   -0.4278
 Psy  91   -0.1371    0.2021   -0.0005   -0.4234
Extr   6   -0.6584    0.0584    0.0581   -0.0637
Extr  11   -0.7479    0.0030    0.0460   -0.0233
Extr  16   -0.6298   -0.0351   -0.0842    0.0599
Extr  20   -0.5488    0.0125   -0.0767    0.1021
Extr  24   -0.7164   -0.0975   -0.0254    0.0351
Extr  45   -0.6743   -0.0272    0.1384   -0.0048
Extr  47   -0.6605   -0.0427    0.0273   -0.0090
Extr  51   -0.6502    0.0208    0.0840   -0.1435
Extr  58   -0.6497   -0.0315   -0.0868    0.2411
Extr  78   -0.6960   -0.0062    0.0414   -0.1088
Extr  90   -0.6026    0.0045   -0.2225    0.0439
Extr  94   -0.7078   -0.0065    0.0593   -0.0904
Neur   3   -0.1298    0.5993   -0.1377   -0.1060
Neur   8    0.0696    0.5907   -0.0987   -0.0481
Neur  17   -0.0024    0.5314   -0.0176   -0.1618
Neur  22   -0.0252    0.5649   -0.0008    0.1537
Neur  26    0.0154    0.6207   -0.0491   -0.0649
Neur  31   -0.0588    0.5355   -0.1720    0.1312
Neur  35    0.0841    0.5549    0.0363   -0.0692
Neur  38    0.0620    0.6863   -0.0192    0.1853
Neur  46   -0.0001    0.6089    0.2438   -0.1948
```

and factor correlations of:

```
           FAC. 1    FAC. 2    FAC. 3    FAC. 4
           ------    ------    ------    ------
 FAC. 1    1.0000    0.0961    0.1773    0.0991
 FAC. 2    0.0961    1.0000   -0.1227   -0.0491
 FAC. 3    0.1773   -0.1227    1.0000    0.1107
 FAC. 4    0.0991   -0.0491    0.1107    1.0000
```

With the Orthogonalised Target as :

```
   Orthogonalised Input Target Matrix:  EPQR Short Form - item numbers referred to 100 item EPQR - MALES - DirObl


                FAC. 1    FAC. 2    FAC. 3    FAC. 4
                ------    ------    ------    ------
 Psy    5    -0.0809   -0.0974   -0.4831    0.1257
 Psy    7    -0.3123   -0.2181   -0.2296    0.0818
 Psy   25    -0.4231    0.0935   -0.1529   -0.0458
 Psy   29    -0.3278    0.0746   -0.3258    0.0476
 Psy   41    -0.1926    0.1536   -0.3704    0.1525
 Psy   48    -0.0767    0.0395   -0.3654   -0.0539
 Psy   54    -0.0219    0.2453   -0.4681    0.1826
 Psy   59    -0.3147   -0.1229   -0.3101    0.0191
 Psy   75    -0.2644    0.0363   -0.2811   -0.0011
 Psy   79    -0.3532    0.0207   -0.3346    0.0201
 Psy   88    -0.3587    0.0684   -0.2868    0.0757
 Psy   91    -0.3664    0.1238   -0.2932   -0.1346
Extr    6    -0.4074   -0.3212    0.0759   -0.3943
Extr   11    -0.4359   -0.4194    0.1272   -0.4128
Extr   16    -0.4040   -0.3707    0.2187   -0.2721
Extr   20    -0.3357   -0.2882    0.2383   -0.2570
Extr   24    -0.4142   -0.4809    0.1856   -0.3191
Extr   45    -0.3066   -0.4205    0.0843   -0.4104
Extr   47    -0.3783   -0.4082    0.1219   -0.3387
Extr   51    -0.4202   -0.3487   -0.0068   -0.3826
Extr   58    -0.3167   -0.3890    0.3737   -0.2931
Extr   78    -0.4538   -0.3927    0.0462   -0.3746
Extr   90    -0.5060   -0.2982    0.2616   -0.2024
Extr   94    -0.4375   -0.4034    0.0565   -0.3909
Neur    3    -0.3699    0.4749    0.0712   -0.2503
Neur    8    -0.1853    0.5687    0.0576   -0.1664
Neur   17    -0.2211    0.4710   -0.0607   -0.2137
Neur   22    -0.0523    0.4660    0.2008   -0.2669
Neur   26    -0.1981    0.5575    0.0389   -0.2324
Neur   31    -0.2041    0.4506    0.2570   -0.1815
Neur   35    -0.0820    0.5254   -0.0232   -0.2122
Neur   38    -0.0212    0.6207    0.2299   -0.2666
```

** Note that the orthogonalised version has now lost it's clean simple structure ...and is almost uninterpretable.

Overall Solution Congruence = **0.97452**

Overall Solution Double-Scaled Euclidean Similarity = **0.96679** [Scaling Minimum = -1.0000 Maximum = 1.0000]

Overall Solution Kernel Distance Smoothed Similarity = **0.94818** [Scaling Smoother value = **0.20000**]

Root Mean Square deviation between the actual input target and original comparison matrix = 0.65994

Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix = 0.91150

Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix = 0.13416

The problem here is that although the overall similarities are useful - when you come to the factor by factor results, it's unclear how to interpret the "factor similarity" - because the factors you and the routine are now looking at are those in the "Orthogonalised Input Target Matrix" - and not the obliquely rotated simple structure factors in the target matrix.

```
 Congruence Coefficients:  between the target and maximally congruent comparison matrix
 ********************************************************************************


           FAC. 1    FAC. 2    FAC. 3    FAC. 4
           ------    ------    ------    ------
 FAC. 1    0.9795    0.0951   -0.1495    0.1364
 FAC. 2    0.1048    0.9871   -0.0170    0.0797
 FAC. 3   -0.1781   -0.0184    0.9484   -0.1182
 FAC. 4    0.1489    0.0790   -0.1084    0.9698
```

If we were to reverse the target and comparison matrix input order - as per example 2, this factor-by-factor matrix is:

```
 Congruence Coefficients:  between the target and maximally congruent comparison matrix
  ********************************************************************************


           FAC. 1    FAC. 2    FAC. 3    FAC. 4
```

```
          ------    ------    ------    ------
FAC. 1    0.9871   -0.0101    0.1090   -0.1214
FAC. 2   -0.0100    0.9759    0.1095   -0.1206
FAC. 3    0.0962    0.0977    0.9509   -0.1321
FAC. 4   -0.1179   -0.1182   -0.1453    0.9682
```

which is close - but not identical - as expected from the data in this example ...

Overall Solution Congruence = 0.97452

Overall Solution Double-Scaled Euclidean Similarity = **0.96655** [Scaling Minimum = -1.0000 Maximum = 1.0000]

Overall Solution Kernel Distance Smoothed Similarity = **0.94771** [Scaling Smoother value = **0.20000**]

Root Mean Square deviation between the actual input target and original comparison matrix = 0.65994

Root Mean Square deviation between the orthogonalised target and orthogonalised comparison matrix = 0.91150

Root Mean Square deviation between the orthogonalised target and now maximally congruent comparison matrix = 0.13416.

Paul Barrett, 30th December, 2019, email: paul@pbarrett.net

Part VI

6 What size Coefficient Indicates Similarity?

6.1 Recommendations

This section is mainly for those using Orthosim-2 for comparison of factor analysis matrices. However, it still has some relevance to those using it for MDS coordinate configural similarity calculation.

Given you observe a congruence coefficient of say 0.85, what do you conclude? Is this good, bad, or indifferent? What about 0.90 or 0.95? Likewise, what about a DSD or KSD or Pearson coefficient of 0.80?

First, let's look at what some investigators/researchers have recommended for a congruence coefficient:

Mulaik, S. (1972) *The foundations of factor analysis*. New York: McGraw-Hill
Recommends **0.85** and above indicates similarity.

Eysenck, H.J. and Eysenck, S.B.G. (1982) Recent advances in the cross-cultural study of personality. In C.D. Spielberger and and J.N. Butcher (eds.) *Advances in Personality Assessment*. Hillsdale: Lawrence Erlbaum.
Recommends **0.95** and above indicates similarity, **0.98** and above indicates essential identity.

Barrett, P. (1986) Factor comparison: An examination of three methods. *Personality and Individual Diff*erences, 7, 3, 327-340.
Recommends **0.90** and above for **mean solution cosine** (overall solution congruence) prior to interpreting factor congruences. Below 0.90 for this overall congruence, the researcher is advised to check the variable-pair congruences for those which are causing a problem. If the congruences fall below certain values, then appropriate warnings will be given at the side of each variable:

If a congruence is less than or equal to **0.4**, then the message "**Trivial Variable Similarity**" is printed alongside the variable congruence.
If a congruence is less than or equal to **0.6** but greater than **0.4**, then the message "**Bad Match**" is printed alongside the variable congruence.
If a congruence is less than or equal to **0.8** but greater than **0.6** then the message "**Poor Match**" is printed alongside the variable congruence.

A minimum factor congruence of **0.80** was suggested as indicative of useful conceptual similarity, with values above these indicating increasing similarity.

Ten Berge, J. M. F. (1986) Rotation to perfect congruence and the cross-validation of component weights across populations. *Multivariate Behavioural Research*, 21, , 41-64.
Recommends **0.85** and above indicates similarity.

Van de Vijver, F.J.R., and Leung, K. (1997) *Methods and Data Analysis for Cross-Cultural Research*. Newbury Park, CA: Sage.
Recommends **0.90** as the lower bound for similarity

Given the Pearson correlation is not recommended for use at all, we are left with the distance coefficients.

The distance coefficients are virtually direct estimates of disparity - hence it is a trivial matter to check the actual factor loading discrepancies against the factor distance coefficients for conceptual significance. Obviously, above **0.90** would indicate a a high degree of similarity. Below this I recommend double-checking the loadings for where the disparity is arising from - an overall "distance" effect or perhaps located most substantially on a few variables out of the total set. Remember also, the KSD coefficient may be designed to substantially penalise any discrepancy, so it's value will reflect how you have chosen to weight discrepancies i.e. 0.90 with a heavy distance penalty is actually indicative of probably greater similarity than a KSD with a lenient penalty function. This is in fact being determined by how you wish to define similarity itself by using such a coefficient. You must go back to the comparison data with a KSD to check where that discrepancy is arising from, and the effect of the weighting on its final value.

Finally, when using raw factor loading comparisons (where no procrustean row normalization has been requested), **never use a congruence coefficient now without confirming the validity of its value using a distance function alongside it**. Example 1 shows why in all its horror! This dual-coefficient approach (covariance and distance) guards against inflated congruences caused by loading monotonicity with magnitude discrepancies between two sets of factor loadings.

Index