

**Normrank Correlations for Testing Associations  
and for use in Latent Variable Models**

Daniel B. Wright

Department of Educational Psychology, Leadership, and Higher Education

University of Nevada, Las Vegas

### Abstract

Pearson's correlation is widely used to test for an association between two variables and also forms the basis of several multivariate statistical procedures including many latent variable models. Spearman's  $\rho$  is a popular alternative. These procedures are compared with ranking the data and then applying the inverse normal transformation, or for short the *normrank* transformation. Using the normrank transformation was more powerful than Pearson's and Spearman's procedures when the distributions have less than normal kurtosis (platykurtic), when the distributions have greater than normal kurtosis (leptokurtic), and when the distribution is skewed. This is examined for testing if there is an association between two variables, identifying the number of factors in an exploratory factor analysis, identifying appropriate loadings in these analyses, and identifying relations among latent variables in structural equation models. R functions and their use are shown.

*Keywords:* robust statistics; latent variable models; structural equation modelling; statistical power

## Normrank Correlations for Testing Associations and for use in Latent Variable Models

Pearson's correlation is widely reported in much of the scientific literature and is the basis for many other statistical procedures. Others calculated correlations before Pearson (e.g., Bravais, Galton), but Pearson showed that the product moment correlation, which bears his name, has some valuable properties for bivariate normal distributions. ?, p. 265, using different notation, defined the correlation between  $x_i$  and  $y_i$  as:

$$r_{xy} = \frac{\sum (x_i - \bar{x}) (y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}} \quad (1)$$

where  $\bar{x}$  and  $\bar{y}$  are the sample means of two variables. It is widely used to examine the association between two variables and correlation matrices are used with factor analysis, structural equation modeling, *etc.* Intervals for these estimates and the associated  $p$ -values are reported in most statistical software. The intervals and  $p$ -values make assumptions about the variables' population distribution (i.e., bivariate normal).

Soon after Pearson's paper, ? devised a method for "characteristics that can *not* be measured quantitatively" (p. 78). It has become known as Spearman's  $\rho$ . His rationale was as follows. If variables cannot be measured quantitatively, meaning is not lost ranking them. Spearman's approach involves ranking the data then applying eqn. 1, though in the days before computers other algorithms were available to make computations easier. It is often used when researchers do not make the bivariate normal assumption or do not want outliers and influential points to have as much impact as with Pearson's correlation. These are different rationale than Spearman's rationale of not losing meaning by ranking, but the procedure is less affected by extreme values than Pearson's  $r$ . Spearman's  $\rho$ , as his version is called, is popular and provides more reliable estimates for distributions with outliers than Pearson's version (e.g., ???). Spearman's procedure involves ranking both variables, assigning the mid-rank to ties, and then using the equation for Pearson's correlation.

Alternative formula for the standard error can be used for constructing intervals and  $p$ -values (see ??).

Numerical transformations can be used to map the bivariate data used in a correlation such that each of the new variables is uniformly distributed (e.g., ?). Ranking is a type of transformation. When there are no ties it results in a discrete uniform distribution. These ranks can be transformed using the inverse of the normal distribution to make their distribution more like this distribution. This is often called the rank based inverse normal transformation (?). For short, here it will be called the *normrank* transformation. ? compared *Student's t* test and the Mann-Whitney Wilcoxon (MWW) test for testing the difference between the center of two samples using this approach. The transformation he used was:

$$new\ x_i = \Phi^{-1} \frac{rank(x_i)}{n + 1}$$

(2)

where  $\Phi^{-1}$  is the inverse of the standard normal function. Consider the data in Table 1. The transformation increases the relative spacing for the first few values but decreases the spacing for the final few values. This is because the first few values are closer to each other than predicted for a tail of the normal distribution but those on the right side of the table are further spread out.

**Table 1**  
*Example values before and after transforming.*

Ten values										
Raw data	-1.00	-0.90	-0.80	-0.60	-0.40	-0.20	0.00	0.50	1.00	2.00
After eqn. 2	-1.34	-0.91	-0.60	-0.35	-0.11	0.11	0.35	0.60	0.91	1.34

Raw data

Transformed data

The transformed variables were then entered into the  $t$ -test formula. van der Waerden called this the  $X$ -test, but it is now usually called the van der Waerden or vdw test. The test is implemented in the R package **PMCMRplus** (?) and has been extended to compare multiple means as an alternative to the traditional oneway ANOVA. ? performed more extensive research comparing this transformation in the  $t$  and  $F$  tests contexts.

The **blom** (named for Blom who suggested  $\delta = 3/8$  in eqn. 3) function from **rcompanion** (?) implements a more general form of this:

$$new\ X_i = \Phi^{-1} \frac{rank(x) - \delta}{n - 2\delta + 1} \quad (3)$$

When  $\delta = 0$  this is the same as eqn. 2, but other values have been proposed . The norm is to have  $\delta \in [-1, +1)$ , where  $[-1, +1)$  is the half-open interval as the upper limit would produce values of  $-\infty$  and  $+\infty$  for the ranks of 1 and  $n$ , respectively (and  $\delta > 1$  produces percentiles outside of  $[0, 1]$  for these values where the inverse normal function is undefined). A simple R function for this is:

```
normrank <- function(x,delta)
  qnorm((rank(x) - delta)/(length(x) - 2 * delta + 1))
```

Similar procedures are available in SAS, SPSS, *etc.*

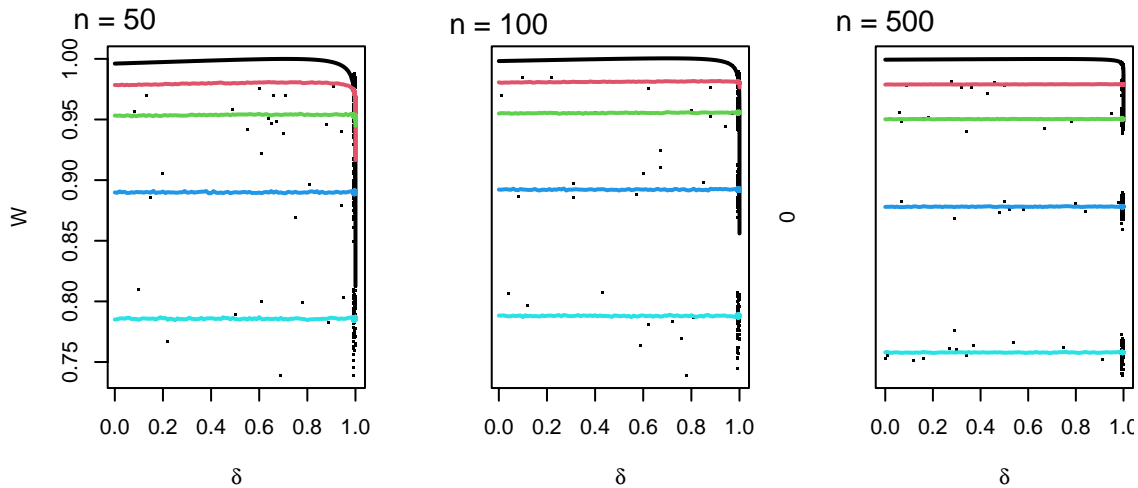
The first simulation in this paper explores different values of  $\delta$  to see which  $\delta$  values, in different circumstances, produce variables that are distributed most similarly to the normal distribution. In subsequent simulations the normranked data are created using this  $\delta$ . The primary purpose of this paper is to show how well this transformation works when applied to two variables, as opposed to a single response variables as examined in ? and ?.

Two issues are important to raise. While this transformation addresses univariate normality, it does not address the assumption of bivariate normality because both variables being normally distributed is a necessary but not a sufficient condition for bivariate

normality (and similarly for multivariate normality). Second, information is lost when ranking the data. This may make it more difficult to interpret the meaning of the resulting correlation as an effect size than Pearson's version. This concern exists also with Spearman's correlation and any rank transformed procedure.

### Simulation 1: Which values of $\delta$ to use?

The `blom` function in `rcompanion` (?) calculates eqn. 3 with  $\delta$  values for 0 (Van der Waerden's),  $\pi/8 \approx 0.393$  (Elving's),  $1/3 \approx 0.333$  (Tukey's),  $3/8 = 0.375$  (Blom's), and  $1/2 = .5$  (the rankit method). Here the values in  $\delta \in [0, 1)$  are examined. Values of  $\delta$  from 0 to .99 in steps of .01, from .99001 to .99999 in steps of .00001, and .999999999, and used. Values very near one are used because  $\delta = 1$  produces  $\infty$  for the rank  $n$  value so there is interest in the transformation's behavior near this threshold. The code for this and the other simulations are in the Appendix 1.



**Figure 1**

*Finding  $W$  values with continuous and discrete scales for different  $\delta$  values, for three sample sizes. The black line is the continuous data, the red line is for the 20-point scale, the green line for the 10-point scale, the darker blue line for the 5-point scale, and the light blue line for the 3-point scale.*

Five levels of discreteness in the data are examined. The first are data drawn from a uniform distribution from  $(0,10)$ , though any continuous distribution of size  $n$  will produce

$n$	Discrete	$\mathbf{W}_{min}$	$\delta_{min}$	$\mathbf{W}_{max}$	$\delta_{max}$
50	Continuous	0.813	1.000	1.000	0.710
50	20pt	0.916	1.000	0.981	0.640
50	10pt	0.945	1.000	0.955	0.770
50	5pt	0.945	0.997	0.892	0.998
50	3pt	0.945	0.992	0.787	0.998
100	Continuous	0.857	1.000	1.000	0.710
100	20pt	0.976	1.000	0.981	0.920
100	10pt	0.955	0.350	0.956	0.991
100	5pt	0.955	1.000	0.894	0.993
100	3pt	0.955	0.992	0.790	0.998
500	Continuous	0.956	1.000	1.000	0.700
500	20pt	0.982	0.040	0.982	1.000
500	10pt	0.957	0.998	0.958	0.995
500	5pt	0.957	0.999	0.896	1.000
500	3pt	0.957	0.996	0.793	0.994

**Table 2**

*Minima and Maxima  $\mathbf{W}$  values and the corresponding  $\delta$ s from the first simulation.*

the same transformed values. The next level uses integers from 1 to 20, created by doubling the first variable, adding .5, and rounding. Analogous approaches were used to create ten point, five point, and three point integer scales, similar to those often used in surveys and personality research. This approach will produce roughly equal numbers of values in each bin. Sample sizes of  $n = 50$ ,  $n = 100$ , and  $n = 500$  were used. Thus, there are  $5 \times 3 = 15$  conditions. The  $\mathbf{W}$  from the Shapiro-Wilk (?) test, as implemented in `shapiro.test` in R using Royston's algorithm (?), is used as a measure to compare distance from normality.  $\mathbf{W}$  is the square of a Pearson correlation so cannot exceed one.

The values are stored, plotted in Figure 1, and the maximum and minimum  $\mathbf{W}$  values, along with their  $\delta$  values shown in Table 2. Note that for the case of continuous data, individual  $\mathbf{W}$  values are not shown because each replication produces the same transformed data and therefore the same  $\mathbf{W}$ . The results show:

1. Little effect in the changes in sample size between  $n = 50$ ,  $n = 100$ , and  $n = 500$ ,
2. The more continuous a variable is, the more closely it can approximate the normal distribution, which is itself continuous,

3. There appears very little differences among the  $\mathbf{W}$ s for the  $\delta$ s  $\in (0, 1]$  for all the discrete variables,
4. There is an effect of  $\delta$  for the continuous variable with the  $\mathbf{W}$  reaching a maximum around .7 (with a maximum  $\mathbf{W}$  rounding to 1.000) and then dropping off when  $\delta$  approaches 1. The dropping off does not happen for the discrete distributions because several values have the extreme bins and therefore their mid-rank is well above 1 or below  $n$ . If samples drawn from populations with fewer values that would be in the extreme bins, they would drop off too as sometimes only one value would be in the extreme bins, and therefore would have a rank of 1 or  $n$ , and a transformed value of it being infinite.
5. The  $\delta$  that produces the highest  $\mathbf{W}$  with discrete values tend to be higher than .7, particularly as the sample size increases.

Here  $\delta = .7$  will be used, though as is clear from the Figure, except when  $\delta \rightarrow 1$  the size of this does not make a large difference. One thousand replications for each  $\delta$  for each condition were used.

### How are data typically distributed?

Simulation is used in this paper to explore the value of the normrank correlation measure. Simulations are done to examine testing for an association, exploratory factor analysis, and structural equation modeling. A selection of the possible data creation scenarios are reported. The aim is to show a selection of uses of this normrank procedure to illustrate its potential, rather than to provide an exhaustive set of possible data creation scenarios.

Before discussing the main simulations reported in this paper it is necessary to examine what types of distributions are typical in education and allied disciplines. The distribution of variables used in correlations varies by discipline and by many other contextual details. The interest here is with kurtosis (for discussion of this statistic, see ??) because of its effect on power (?). There are several types of kurtosis based on the fourth



moment and more robust alternatives (e.g., ?). These are reviewed by ?. Type 3 kurtosis is the default of the function `kurtosis` found in the **e1071** package (?) and is reported here.

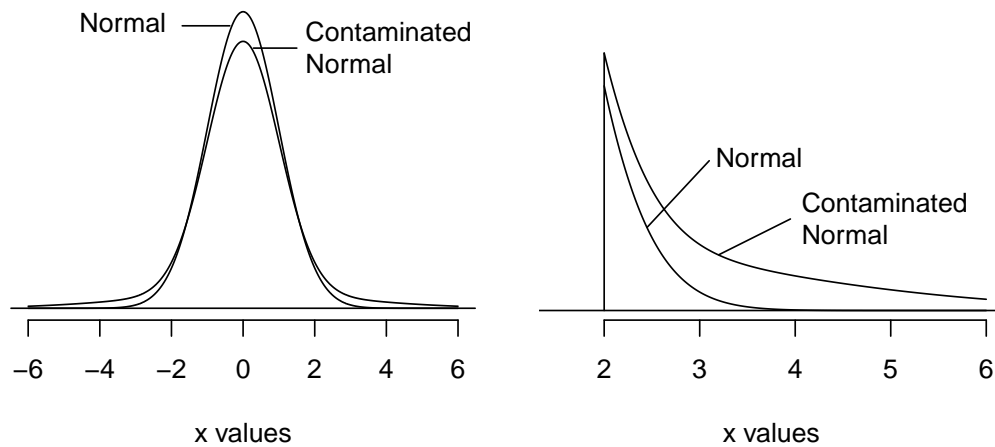
? requested data from several education sources, test publishers, and authors of psychology and education journal articles. He analyzed 440 separate distributions with respect to kurtosis. He classified their kurtosis as: similar to the uniform distribution (3.2% of his sample), less than the normal distribution (14.8%), similar to the normal distribution (15.8%), similar to a moderate contaminated normal distribution (17.7%), similar to an extreme contaminated normal distribution (32.5%), and similar to a double exponential distribution (16.6%). He defined a moderate contaminated distribution as 95% drawn from a normal distribution with  $\sigma = 1$  and 5% drawn from a normal distribution with  $\sigma = 2$ . Extreme contaminated normal was defined as 85% drawn from a normal distribution with  $\sigma = 1$  and 15% drawn from a normal distribution with  $\sigma = 3$ . Thus, about half of the distributions he examined had kurtosis at least that of what he called the extreme contaminated normal distribution. He found approximately 18% to have kurtosis less than normal.

Paraphrasing Poincaré, everybody believes the normal distribution assumption is justified: the mathematicians because they think the scientists say all their data are normally distributed; the scientists because they think the mathematicians say it is justified because of the central limit theorem. ? titled his paper “The unicorn, the normal curve, and other improbably creatures” to stress that real data are seldom normally distributed. Thus, scientists do not say that all their data are usually normally distributed. ? showed that non-normally distributed data do affect statistical results, countering the idea that the central limit theory makes this assumption justified in all circumstances. Tukey’s advice was to assume that “all assumptions are wrong” (?, p. 72), but then to explore how problematic it is to make them and what alternatives exist. One of the main goals of this paper is to examine the costs of assuming data are normally distributed and discuss an alternative.

Four distributions are considered here to illustrate how the three correlations perform: the uniform distribution, the normal distribution, a contaminated normal distribution with 15% with  $\sigma = 3$ , and the  $\chi^2$  with  $df = 1$ , which is equivalent to the square of a normal distribution. The normal and contaminated normal distributions are shown in Figure 2. They can appear very similar in histograms, but the tails of the contaminated normal distribution are larger (right panel). Tukey showed that this can have a large impact on the power of the tests as the standard deviations can become much larger. The kurtosis for these four distributions are: uniform equals 1.80, normal equals 0,  $\chi^2_1$  equals 12, and the contaminated normal used equals approximately 5.06. The  $\chi^2_1$  distribution has skewness of  $\sqrt{8} \approx 2.83$ . The others are symmetric. An example of how to perform a normrank correlation is in the first section of Appendix 2.

**Figure 2**

*The normal distribution ( $\mu = 0, \sigma = 1$ ) and a contaminated normal distribution that includes 15% with  $\sigma = 3$ . The right panel focuses on the tails with  $x > 2$ . This makes the heavier tail of the contaminated distribution clearer.*



### Simulation 2: Testing for an Association

Pearson's correlation, Spearman's correlation, and the normrank correlation are compared for whether they maintain the stated Type 1 error rate, here 5%, and for their

power to detect an association. Other tests of association exist (e.g., polychoric correlations), but the desire was to compare these because they all use the same procedure after transforming the original data (with no transformation, ranking, or with eqn. 3 using  $\delta = .7$ ) and the popularity of the first two procedures. A concern for Pearson's approach is that it is greatly influenced by extreme values (e.g., ?). The ranking transformation used with Spearman's correlation directly addresses this so that all values are equally spaced, provided that there are no ties (ties, and discrete variables generally, are not considered in this paper). Compared with ranking, eqn. 3 changes this so that more extreme rank values are spaced more than less extreme rank values. Whether these extreme values are spaced further apart than the original data depends on the distribution of the original data (see Table 1).

First consider the performance of these measures when there is no association. Sample sizes of 40, 60, and 100 were used. The distributions—uniform, normal, contaminated normal,  $\chi^2_1$ —were the same for both variables. The  $\chi^2$  distribution with  $df = 1$ , which is equivalent to squaring a normally distributed variable with  $\mu = 0$  and  $\sigma = 1$ , was included so that one of the variables is skewed (skewness is  $\sqrt{8} \approx 2.83$ ). There were 10,000 replications per condition for the remaining simulations reported in this paper. The R code for these is in the Appendix 1 and can be adapted for different sample sizes, distributions, and other parameters. The  $p$ -values are found for the Pearson, Spearman, and normrank correlations using R's `cor.test` function. The nominal  $\alpha$  is .05, so the proportion of  $p < .05$  should be near this value. The proportions of  $p < .05$  results and the 95% confidence intervals for these proportions, found with the Wilson method, are reported. Because the number of replications is large, the widths of these intervals are small. Differences of .01 between the correlation procedures are large enough that they are discussed and differences greater than .02 may be considered substantial in some contexts.

Table 3 shows for each procedure, for each distribution and for each sample size, the proportion of  $p$ -values less than .05 and the confidence intervals of these proportions. All

are near the nominal level of .05. None of the individual cell proportions is greater than .01 away from the nominal level. Therefore, it is concluded that all are adequate at controlling Type 1 error.

**Table 3**

*Proportion of replications where  $p < .05$  for the different distributions, sample sizes, and correlation measures when there is no association (i.e., when the null hypothesis is true).*

	Correlation Measures					
	Pearson		Spearman		normrank	
	Prop.	95% CI	Prop.	95% CI	Prop.	95% CI
n = 40						
Uniform	.048	(.044, .053)	.048	(.044, .053)	.047	(.043, .051)
Normal	.052	(.048, .056)	.053	(.049, .057)	.051	(.047, .055)
Contaminated	.055	(.051, .059)	.053	(.049, .058)	.053	(.049, .058)
$\chi^2_1$	.049	(.045, .053)	.054	(.050, .058)	.051	(.046, .055)
n = 60						
Uniform	.052	(.048, .057)	.052	(.048, .057)	.053	(.049, .058)
Normal	.050	(.046, .055)	.050	(.046, .054)	.050	(.046, .054)
Contaminated	.055	(.051, .060)	.054	(.050, .059)	.053	(.048, .057)
$\chi^2_1$	.046	(.042, .050)	.053	(.048, .057)	.050	(.046, .055)
n = 100						
Uniform	.049	(.045, .054)	.047	(.043, .052)	.046	(.042, .050)
Normal	.052	(.048, .056)	.049	(.045, .054)	.050	(.046, .055)
Contaminated	.051	(.047, .056)	.052	(.048, .057)	.052	(.048, .057)
$\chi^2_1$	.049	(.045, .054)	.053	(.049, .058)	.052	(.048, .057)
95% confidence intervals found with Wilson's method.						
Ten thousand replications per cell.						

This simulation was repeated, but with the null hypothesis being false. The variables were created using the three distributions but for one of the variables .32 times the other was added. This results in a Pearson correlation of approximately  $r = .30$ , what ? called a medium sized effect, for the normally distributed variables. The correlations are slightly different for the other distributions and other measures. The primary interest here is whether the different statistical procedures vary in how well they detect an association for the same data, not differences across the distributions.

Table 4 shows the proportion of the replications where the null hypothesis was correctly rejected for each procedure, for each distribution, and for each sample size. As expected, these proportions increase with sample size (i.e., the power increases with the

sample size). For the normal distribution, Pearson's correlation correctly rejects  $H_o$  about 1% more often than the normrank correlation and about 4-5% more than Spearman's correlation. Thus, in this situation, Spearman's is the least powerful and Pearson's is the most powerful. With both the uniform and contaminated normal distributions, the normrank procedure had about a 3-4% advantage over Pearson's with Spearman's performing between the two. Based on Micceri's (?) findings, where most distributions he analyzed had greater than normal kurtosis and a small proportion less than normal, it is likely that education and other social science researchers will face circumstances where the normrank is more powerful than Pearson's procedure. In none of the situations examined was Spearman's procedure the most powerful procedure. It is recommended not to use Spearman's correlation and it will not be further examined in this paper.

Table 4

Proportion of replications where  $p < .05$  for the different distributions, sample sizes, and correlation measures when there is an association (i.e., when the null hypothesis is false).

	Correlation Measures					
	Pearson		Spearman		normrank	
	Prop.	95% CI	Prop.	95% CI	Prop.	95% CI
n = 40						
Uniform	.497	(.487, .507)	.448	(.438, .458)	.541	(.532, .551)
Normal	.493	(.484, .503)	.446	(.436, .456)	.473	(.463, .483)
Contaminated	.497	(.487, .506)	.502	(.492, .511)	.524	(.514, .533)
$\chi^2_1$	.488	(.478, .498)	.690	(.681, .699)	.698	(.689, .707)
n = 60						
Uniform	.669	(.660, .679)	.611	(.602, .621)	.736	(.727, .745)
Normal	.669	(.660, .678)	.620	(.610, .629)	.647	(.637, .656)
Contaminated	.660	(.651, .670)	.688	(.679, .697)	.700	(.691, .709)
$\chi^2_1$	.632	(.622, .641)	.861	(.854, .867)	.871	(.864, .877)
n = 100						
Uniform	.882	(.876, .888)	.836	(.828, .843)	.933	(.928, .938)
Normal	.879	(.872, .885)	.839	(.832, .846)	.867	(.860, .874)
Contaminated	.864	(.857, .870)	.897	(.891, .903)	.904	(.898, .910)
$\chi^2_1$	.842	(.835, .849)	.978	(.975, .980)	.982	(.979, .984)

95% confidence intervals found with Wilson's method.  
Ten thousand replications per cell.

### Simulation 3: Exploratory Factor Analysis (EFA)

In this simulation Pearson's correlation and the normrank correlation are compared for exploratory factor analysis (EFA) when there are no underlying factors and when there are 1 to 4 underlying factors with twelve observed variables. Spearman's is not included as it performed worse than others for all distributions tested in the previous simulation.

EFA is one of the most used forms of latent variable model particularly within the educational and social sciences. It is usually the first latent variable model that students are taught. If you have some number of observed variables and believe that this observed set is caused by a smaller number of latent variables, but you do not have particular relationships hypothesized, EFA can be an appropriate tool. EFA produces two main results. First, it can suggest how many latent variables are appropriate for modeling the data. This is often done using a related technique: calculating the eigenvalues of the correlation matrix. The second result is that it shows which observed variables are related to the same latent variable as others. There are several sources for EFA, for example, in increasing level of assumed mathematical knowledge: [? ? ? ?](#), and [?](#).

The simulation in this section will examine whether using a Pearson correlation matrix or a matrix composed using the normrank transformation is better able both to determine the correct number of latent variables and to identify the loadings correctly. [?](#), see also [? ?](#) review different methods to determine the number of factors. Based on this, the empirical Kaiser criterion is used ([?](#)). It involves comparing the observed eigenvalues with those observed from a random data matrix. It is implemented in the **EFA.dimensions** package ([?](#)). To estimate whether the pattern of factor loadings is accurate, factor analysis was conducted assuming the correct number of factors. The highest loading for each observed variable was recorded. For the complete pattern of loadings to be deemed accurate, all observed variables which were influenced by the same latent variable during data creation had to have their highest loading for the same latent variable. This was done with the default settings of the R function `factanal`.

Only a single sample size is used here. There is no agreed upon minimum sample size for EFA (??), but several rules of thumb exist. ? say “with high commonalities, a small number of factors, and a relatively large number of indicators per factor,  $N$ s of 100 or or less yield good recovery of factors” (p. 207). A sample size of 100 will be used with twelve observed variables since it is near the minimum that might be considered adequate (though larger samples are recommended). Each observed variable will be created as:  $observed_i = latent_i + x \cdot noise_i$ , where  $x$  is a scalar to control the amount of noise (see below).

The latent variables and the noise variables are each drawn from one of several contaminated normal distributions. Because ? found the preponderance of the variables he examined to have greater than normal kurtosis, this simulation will focus on distributions with varying amounts of contamination for the contaminated normal distribution. In addition, results for the  $\chi_1^2$  distribution are presented for comparison to show a skewed distribution.

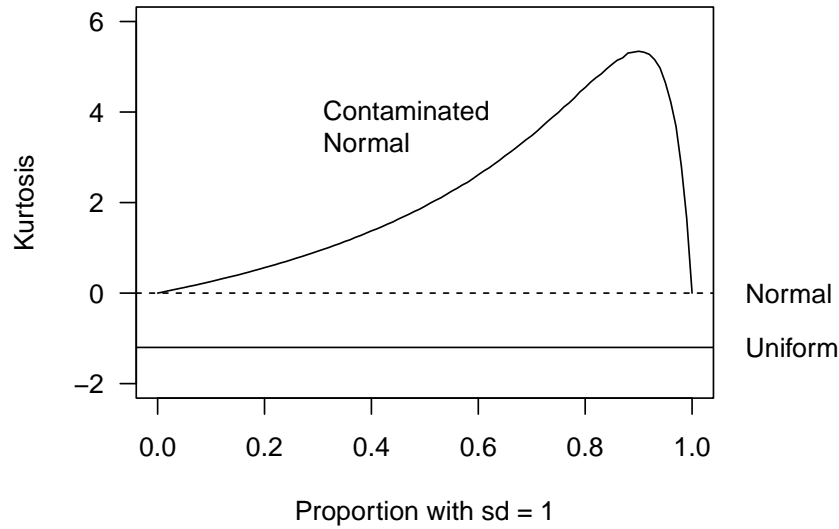
Contamination for contaminated normal distributions can be created in different ways. Here the variability of the two distributions are fixed at  $\sigma_1 = 1$  and  $\sigma_2 = 3$  and the proportion from each of these distributions is varied between 0 and 1, with the endpoints being normal distributions. This is done in 101 steps of .01. The kurtosis for these different contaminated normal distributions is shown in Figure 3. The maximum kurtosis is when about 10% is drawn from a normal distribution with  $\sigma = 3$ . The uniform distribution is presented for comparison.

Different amounts of noise are used for the different numbers of latent variables in order to prevent results being at ceiling (correctly identifying that there are four latent variables is more difficult than identifying that there is one). The scalars for noise are  $x = 2.5$ ,  $x = 2$ ,  $x = 1.5$ , and  $x = 1.2$  for one, two, three, and four latent variables respectively.

Figures 4 and 5 show that when the data are all produced by non-contaminated

**Figure 3**

Kurtosis for normal distributions with  $\sigma = 1$  for proportions from 0 to 1 with the remainder being drawn from normal distributions with  $\sigma = 3$ , as well as the kurtosis for the uniform distribution.

**Table 5**

Results using  $\chi^2_1$  distributions for exploratory factor analyses with the data created from different numbers of latent variables. There were 10,000 replications.

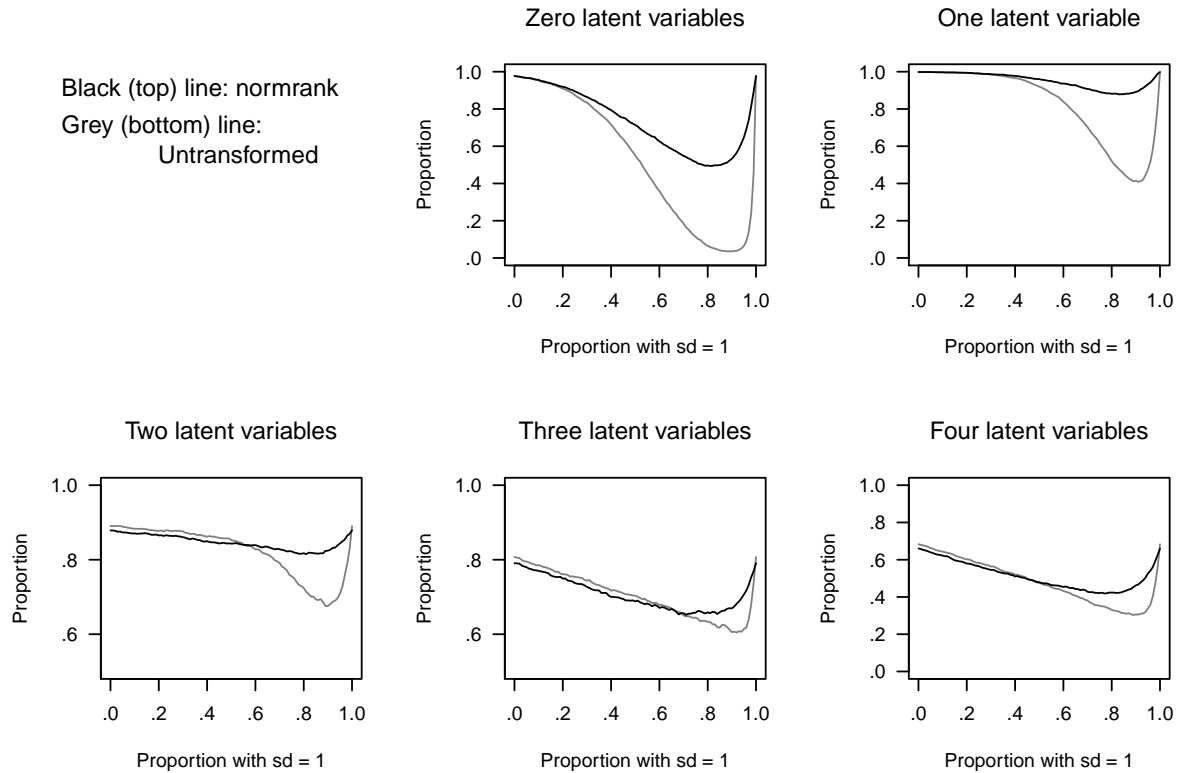
		Number of True Factors				
		0	1	2	3	4
Finding the correct number of factors						
Pearson	Proportion	.963	.747	.777	.440	.165
	95% CI	(.959, .966)	(.739, .756)	(.768, .785)	(.431, .450)	(.158, .173)
normrank	Proportion	.974	.994	.988	.777	.327
	95% CI	(.971, .977)	(.992, .995)	(.986, .990)	(.769, .786)	(.318, .336)
Finding the correct loadings						
Pearson	Proportion			.791	.727	.691
	95% CI			(.783, .799)	(.718, .735)	(.682, .700)
normrank	Proportion			.973	.936	.896
	95% CI			(.969, .976)	(.931, .941)	(.889, .901)

normal distributions (mixtures of 0 and 1), Pearson's and the normrank correlation perform similarly for both correctly identifying the number of factors and identifying the correct pattern of loadings. They perform similarly until the distribution has less than



**Figure 4**

*Proportion of replications identifying the correct number of latent factors. Note that the y-axes are different for the five plots. The proportion from the population with  $\sigma = 3$  increased from 0 to 1 in steps of .01, with 10,000 replications at each step.*



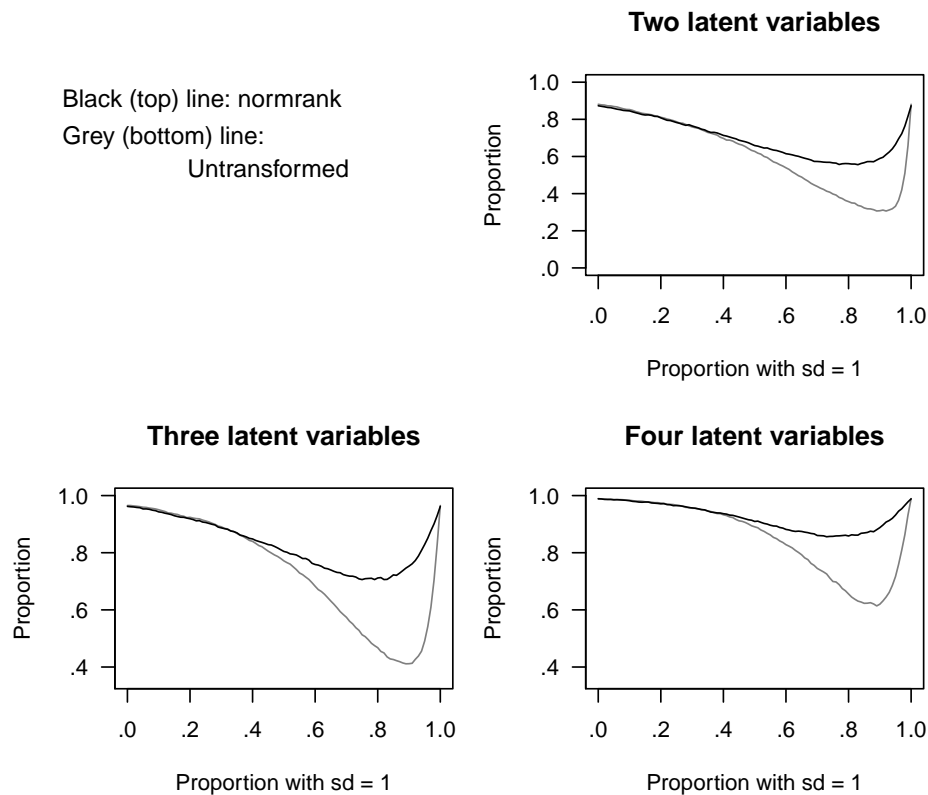
about 50% drawn from the one with the larger standard deviation. This is also where the kurtosis becomes noticeable larger (Figure 3). For these distributions the normrank procedure was more accurate both for identifying the correct number of factors and for identifying the correct pattern of factor loadings.

Table 5 shows the results when the data are distributed  $\chi_1^2$ . When there is more than one factor used to create the data, the normrank correlation is more accurate than Pearson's correlation both for identifying the number of factors and the factor loading pattern.

There are many different parameters that could be used to construct data for EFA and different ways to assess the accuracy of the results. The results presented here are just to illustrate the use of the two correlation procedures. For these, and consistent with the

**Figure 5**

*Proportion of time identifying the correct loadings (i.e., the highest loading being the same as the others influenced by the same factor). The proportion from the population with  $\sigma = 3$  increased from 0 to 1 in steps of .01, with 10,000 replications at each step. Note that the y-axis scales are different.*



correlation simulation, when the data derive from normal distributions the Pearson's and the normrank correlation perform similarly, but for non-normal distributions, both those with less than normal kurtosis (the uniform distributions) and those with greater than normal kurtosis (the contaminated normal distributions), the normrank performed better. Given Micceri's (1989) finding that most of the distributions he analyzed in psychology and education were not similar to the normal distribution, and most of these had greater than normal kurtosis, the recommendation here is that the normrank correlation should be preferred over Pearson's correlation when conducting EFA for many psychology and education data sets.

### Simulation 4: Structural Equation Modeling (SEM)

SEM can take numerous forms. It allows the researcher to hypothesize relationships among latent variables. However, it is a complex analysis that should be used cautiously.

When we come to models for relationships between latent variables we have reached a point where so much has to be assumed that one might justly conclude that the limits of scientific usefulness have been reached, if not exceeded.

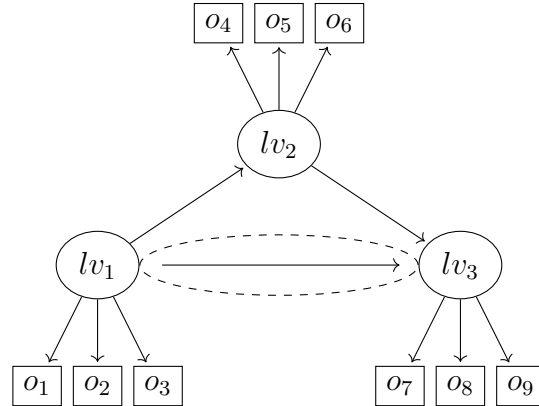
?, p. 228

A good applied introduction on SEM using R is ?, with more coverage in ?. The **lavaan** manual ?, one of the main R SEM packages, is a good resource. Here only a single example is shown in order to show how the normrank correlation compares with Pearson's correlation using three distributions (uniform, normal, contaminated normal) for a relatively simple SEM model. A single situation is used to explore if the findings are consistent with the earlier simulations. There are numerous variations in SEM models that could be explored further. As with the other simulations, within each data creation scenario all the random variables will have the same distributions: uniform (range 0 to 1), normal ( $\mu = 0$ ,  $\sigma = 1$ ), and contaminated normal (85% drawn from a normal with  $\mu = 0$  and  $\sigma = 1$  and 15% drawn from a normal with  $\mu = 0$  and  $\sigma = 3$ ).

The data will be created from the model depicted in Figure 6. There are three latent variables,  $lv_1 - lv_3$ , each influencing three observed variables, labeled  $o_1 - o_9$ .  $lv_1$  influences  $lv_2$ , and  $lv_2$  influences  $lv_3$ . For half of the simulations  $lv_1$  also influences  $lv_3$ . The statistical models compare the data creation models with and without this effect and evaluated for whether this effect is detected at  $p < .05$ . Given the nature of SEM modeling, sometimes models fail to converge. Often alternative estimation procedures and different control settings can be used to achieve convergence. Here the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm is used, but some errors persisted and are treated as missing. Only 0.12% of the 20,000 replications failed to produce a result.

**Figure 6**

*Causal model used to create the data for Structural Equation Modeling (SEM) simulation. The effect corresponding with the arrow surrounded by the dashed ellipse is included for half the samples.*

**Table 6**

*Proportion of replications rejecting the hypothesis that there is a direct effect from  $lv_1$  to  $lv_3$  (the effect in the dashed ellipse in Figure 6). For half of the simulations these are incorrect decisions (top half of the table) and should be near the  $\alpha = .05$  level. For half they are the correct decisions (the bottom half). There were 10,000 replications for each condition.*

		uniform	Normal	Contaminated
Null hypothesis true ( $lv_1 \rightarrow lv_3$ not present)				
Pearson's	Proportion	.170	.080	.240
	95% CI	(.162, .177)	(.075, .086)	(.232, .249)
normrank	Proportion	.150	.082	.219
	95% CI	(.143, .157)	(.077, .088)	(.211, .227)
Null hypothesis false ( $lv_1 \rightarrow lv_3$ present)				
Pearson's	Proportion	.484	.495	.351
	95% CI	(.474, .494)	(.485, .505)	(.341, .360)
normrank	Proportion	.496	.494	.418
	95% CI	(.487, .506)	(.484, .504)	(.408, .428)

The primary SEM results can be summarized as: Pearson's and the normrank produce similar findings when the distributions are uniform or normal, but the normrank approach performs better both in terms for not rejecting the true null (by a couple of percent) and correctly rejecting a false null hypothesis (by about seven percent) when the distributions have high kurtosis. Large differences were found for the different distributions for incorrectly rejecting a true null (i.e., the Type 1 error proportions). In the uniform conditions the true null was incorrectly rejected less than the nominal level of .05, while in

the normal conditions it was rejected slightly more than the nominal level, and in the contaminated normal conditions it was rejected much more. These differences highlight the importance of considering how the distributions underlying the data (real and simulate) can affect statistical models.

### Summary

Pearson's product moment correlation is one of the most used statistics in all of science. It is highly influenced by extreme values which makes this procedure less reliable for some distributions. Spearman's rank correlation is a popular alternative when there are extreme outliers and the analyst wants to limit their impact. The normrank correlation, using what is sometimes called the rank based inverse normal transformation, results in data more closely resembling a normal distribution.

Four simulations were conducted. The first was to show which variation of the normrank transformation to use. For continuous distributions  $\delta = .7$  appeared to perform well. This was used in the remaining simulations. The second simulation examined how well Pearson's, Spearman's, and the normrank correlation performed detecting an association. Spearman's was never the best of these three. The recommendation is not to use it. Pearson's was more powerful than the normrank correlation with the normally distributed data, but the difference was fairly small. The normrank correlation was more powerful than Pearson's correlation when the distribution had sub- and super-normal kurtosis. Because ? found most correlations in psychology and education had greater than normal kurtosis, the normrank correlation is likely to be more powerful and should be considered. However, it is important to stress that Pearson's approach was better for normally distributed data, as has been shown in other contexts (e.g., ?), using the normrank transformation will not always produce more powerful statistical tests. A further reservation for all procedures involving ranking is that information about the original variable metric is lost.

The final two simulations compared the use of Pearson's correlation with the

normrank correlation in factor analysis and structural equation modeling. The results were similar to the second simulation. When the data are drawn for normal distributions, the procedures perform similarly, but when they are drawn from non-normal distributions, with kurtosis much different from the normal distribution, the normrank performed better. Overall,

It is concluded, therefore that:

1. Spearman's procedure should not be used.
2. The normrank should be considered unless the researcher has strong reasons to believe their data are drawn for near-normal distributions, but researchers may also wish to consider numeric transformations.

It is important to stress that this paper only addresses one alternative to Pearson's correlation on untransformed variables. The paper shows that the normrank transformation can improve the subsequent statistics, but there are other approaches. First, if there is a particular substantive model that is consistent with a particular transformation (e.g., count data, which according to ?, may profit from the  $\ln$  transformation) this should be considered. Second, the Box-Cox transformation (?) usually makes data more similar to the normal distribution and is often used. If you have a particular model, whether linear or otherwise, about the relationship between the variables no transformation involving ranking should be used and the applicable model should be evaluated. If the worry is about the impact of extreme values more robust loss functions could be used (e.g., ?) and if the worry is about not meeting the distributional assumptions bootstrapping could be used (e.g., ??). In short, there are many alternatives to conducting Pearson's correlation on untransformed variables, and in some situations the normrank transformation data should be considered.

In line with the goals of this journal to have access to as much evidence as plausible (see ?), the R code to reproduce the simulations is in Appendix 1. Further, the final submitted document, in full, is on the author's github page

(<https://github.com/dbrookswr/normrank/blob/main/README.md>). It includes L<sup>A</sup>T<sub>E</sub>X and R, both freely available, knitted together using `knitr` (?).

## Appendix 1: Code for Simulations

The final submitted document is available as a reproducible `knitr` document (?) at the author's github page (\*\*). The code for the simulations is shown in the Appendix.

### Simulation 1

Below is the code for when  $n = 50$  to find values for  $\delta$  that produce variables that are distributed similarly to the normal distribution. The simulations for  $n = 100$  and  $n = 200$  are identical except that the sample size is changed.

```
normrank <- function(x,delta)
  qnorm((rank(x) - delta)/(length(x) - 2 * delta + 1))
set.seed(5859)
reps <- k <- 1000
vals2 <- matrix(nrow=5*k*length(seqvals),ncol=5) # trialno, n, discreteness, seqvals, s
count <- 0
for (ww in 1:5){
  for (i in 1:reps){
    for (j in 1:length(seqvals)) {
      count <- count + 1
      n <- 50
      x <- runif(n,0,10)
      if(ww == 2) x <- round(2*x+.5)
      if(ww == 3) x <- round(x +.5)
      if(ww == 4) x <- round(x/2 +.5)
      if(ww == 5) x <- round(3*x/10 +.5)
      newx <- normrank(x,delta=seqvals[j])
      w <- shapiro.test(newx)$statistic
      vals2[count,] <- c(count,n,ww,seqvals[j],w)
    } } }
```

### Simulation 2

These are the functions used to make the contaminated normal distributions and calculate the different correlations:

```

rmixnorm <- function(n,mix = .85, wide = 3) {
  if (abs(round(n*mix)) != n*mix)
    warning("Sample does not divide by mix. Will get close.")
  g1 <- abs(round(n*mix))
  sample(c(rnorm(g1),rnorm(n-g1,sd=wide)))
}

normcorr <- function(x,y) cor.test(normrank(x,.7),normrank(y,.7))
sigbw <- function(r,n) sqrt((1+r^2/2)/(n-3))
cibw <- function(r,n)
  tanh(atanh(r) + c(-1,1)*sigbw(r,n) * qt(.975,n-2))

corr3 <- function(x,y,asvec=TRUE){
  pear <- cor.test(x,y)
  spear <- cor.test(x,y,method="spearman")
  spearmanCI <- cibw(spear$estimate,length(x))
  norm <- normcorr(x,y)
  ifelse(asvec,
    pcis <- c(pear$p.value,pear$conf.int[1],pear$conf.int[2],
              spear$p.value,spearmanCI[1],spearmanCI[2],
              norm$p.value, norm$conf.int[1], norm$conf.int[2]),
    {pcis <- matrix(c(pear$p.value,pear$conf.int[1],pear$conf.int[2],
                     spear$p.value,spearmanCI[1],spearmanCI[2],
                     norm$p.value, norm$conf.int[1], norm$conf.int[2]),ncol=4)
      rownames(pcis) <- c("p","lb","ub")
      colnames(pcis) <- c("Pearson","Spearman","normCorr")})
  return(pcis)
}

```

This is the code used to create the data for the correlation simulation with  $H_0$  true and  $n = 40$ . The  $n$  is changed for the other two sample sizes.

```

k <- 10000
n <- 40
vals1_40 <- array(dim=c(k,10,length(dists)),
  dimnames =
    list(c(paste0("replics",1:k)),
         c("i","Pearp","Pearlb","Pearub",
           "Spearp","Spearlb","Spearub",
           "normp","normlb","normub"),
         c("unif","norm","mixnorm","chi1")))
for (j in 1:length(dists))
  for (i in 1:k){

```



```

set.seed(8184+i)
x <- dists[[j]](n)
y <- dists[[j]](n) # + part of x if power
cc <- corr3(x,y)
vals1_40[i,,j] <- c(i,corr3(x,y))
}

```

### Simulation 3

The data were created for the EFA with the following functions, one each depending on the number of latent variables.

```

create0data <- function(n,mix,sigma=3,noise=1){
  ov1 <- noise*rcnorm(n,mix,sigma=sigma)
  ov2 <- noise*rcnorm(n,mix,sigma=sigma)
  ov3 <- noise*rcnorm(n,mix,sigma=sigma)
  ov4 <- noise*rcnorm(n,mix,sigma=sigma)
  ov5 <- noise*rcnorm(n,mix,sigma=sigma)
  ov6 <- noise*rcnorm(n,mix,sigma=sigma)
  ov7 <- noise*rcnorm(n,mix,sigma=sigma)
  ov8 <- noise*rcnorm(n,mix,sigma=sigma)
  ov9 <- noise*rcnorm(n,mix,sigma=sigma)
  ov10 <- noise*rcnorm(n,mix,sigma=sigma)
  ov11 <- noise*rcnorm(n,mix,sigma=sigma)
  ov12 <- noise*rcnorm(n,mix,sigma=sigma)
  return(cbind(ov1,ov2,ov3,ov4,ov5,ov6,ov7,ov8,ov9,ov10,ov11,ov12))
}

create1data <- function(n,mix,sigma=3,noise=1){
  lv1 <- rcnorm(n,mix,sigma=sigma)
  ov1 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov2 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov3 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov4 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov5 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov6 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov7 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov8 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov9 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov10 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov11 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov12 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
}

```

```

    return(cbind(ov1,ov2,ov3,ov4,ov5,ov6,ov7,ov8,ov9,ov10,ov11,ov12))
  }

create2data <- function(n,mix,sigma=3,noise=1){
  lv1 <- rcnorm(n,mix,sigma=sigma)
  lv2 <- rcnorm(n,mix,sigma=sigma)
  ov1 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov2 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov3 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov4 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov5 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov6 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov7 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov8 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov9 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov10 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov11 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov12 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  return(cbind(ov1,ov2,ov3,ov4,ov5,ov6,ov7,ov8,ov9,ov10,ov11,ov12))
}

create3data <- function(n,mix,sigma=3,noise=1){
  lv1 <- rcnorm(n,mix,sigma=sigma)
  lv2 <- rcnorm(n,mix,sigma=sigma)
  lv3 <- rcnorm(n,mix,sigma=sigma)
  ov1 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov2 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov3 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov4 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
  ov5 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov6 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov7 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov8 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
  ov9 <- lv3 + noise*rcnorm(n,mix,sigma=sigma)
  ov10 <- lv3 + noise*rcnorm(n,mix,sigma=sigma)
  ov11 <- lv3 + noise*rcnorm(n,mix,sigma=sigma)
  ov12 <- lv3 + noise*rcnorm(n,mix,sigma=sigma)
  return(cbind(ov1,ov2,ov3,ov4,ov5,ov6,ov7,ov8,ov9,ov10,ov11,ov12))
}

create4data <- function(n,mix,sigma=3,noise=1){
  lv1 <- rcnorm(n,mix,sigma=sigma)

```

```

lv2 <- rcnorm(n,mix,sigma=sigma)
lv3 <- rcnorm(n,mix,sigma=sigma)
lv4 <- rcnorm(n,mix,sigma=sigma)
ov1 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
ov2 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
ov3 <- lv1 + noise*rcnorm(n,mix,sigma=sigma)
ov4 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
ov5 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
ov6 <- lv2 + noise*rcnorm(n,mix,sigma=sigma)
ov7 <- lv3 + noise*rcnorm(n,mix,sigma=sigma)
ov8 <- lv3 + noise*rcnorm(n,mix,sigma=sigma)
ov9 <- lv3 + noise*rcnorm(n,mix,sigma=sigma)
ov10 <- lv4 + noise*rcnorm(n,mix,sigma=sigma)
ov11 <- lv4 + noise*rcnorm(n,mix,sigma=sigma)
ov12 <- lv4 + noise*rcnorm(n,mix,sigma=sigma)
return(cbind(ov1,ov2,ov3,ov4,ov5,ov6,ov7,ov8,ov9,ov10,ov11,ov12))
}

```

The simulation for the situation with no latent variables was conducted with the following.

The other simulations used the other functions, from above, to create the data.

```

set.seed(3940)
k <- 10000
n <- 100
vals0 <- array(dim=c(k,5,length(mixvals)),
  dimnames =
    list(c(paste0("replics",1:k)),
         c("i","nf1EMPKC","nf2EMPKC","load1","load2"),
         paste0("mix",mixvals)))
for (j in 1:length(mixvals)){
  for (i in 1:k){
    set.seed(2712+i)
    x <- create0data(n,mixvals[j])
    vals0[i,,j] <- c(i,fa(x,fa=0))
  }
}

```

## Simulation 4

The following creates the SEM data.

```

makesemu <- function(n=200,noise=1,effect=TRUE){
  lv1 <- rchisq(n,df=1)
  o1 <- lv1 + noise*rchisq(n,df=1)
  o2 <- lv1 + noise*rchisq(n,df=1)
  o3 <- lv1 + noise*rchisq(n,df=1)
  lv2 <- scale(lv1 + noise*rchisq(n,df=1))
  o4 <- lv2 + noise*rchisq(n,df=1)
  o5 <- lv2 + noise*rchisq(n,df=1)
  o6 <- lv2 + noise*rchisq(n,df=1)
  ifelse(effect,
    lv3 <- scale(lv1/2 + lv2/2 + noise*rchisq(n,df=1)),
    lv3 <- scale(lv2 + noise*rchisq(n,df=1)))
  o7 <- lv3 + noise*rchisq(n,df=1)
  o8 <- lv3 + noise*rchisq(n,df=1)
  o9 <- lv3 + noise*rchisq(n,df=1)
  x <- cbind(o1,o2,o3,o4,o5,o6,o7,o8,o9)
  colnames(x) <- paste0("o",1:9)
  return(x)
}

makesemn <- function(n=200,noise=1,effect=TRUE){
  lv1 <- rnorm(n)
  o1 <- lv1 + noise*rnorm(n)
  o2 <- lv1 + noise*rnorm(n)
  o3 <- lv1 + noise*rnorm(n)
  lv2 <- scale(lv1 + noise*rnorm(n))
  o4 <- lv2 + noise*rnorm(n)
  o5 <- lv2 + noise*rnorm(n)
  o6 <- lv2 + noise*rnorm(n)
  ifelse(effect,
    lv3 <- scale(lv1/2 + lv2/2 + noise*rnorm(n)),
    lv3 <- scale(lv2 + noise*rnorm(n)))
  o7 <- lv3 + noise*rnorm(n)
  o8 <- lv3 + noise*rnorm(n)
  o9 <- lv3 + noise*rnorm(n)
  x <- cbind(o1,o2,o3,o4,o5,o6,o7,o8,o9)
  colnames(x) <- paste0("o",1:9)
  return(x)
}

makesemcn <- function(n=200,noise=1,effect=TRUE){
  lv1 <- rcnorm(n)
  o1 <- lv1 + noise*rcnorm(n,mix=.85)
  o2 <- lv1 + noise*rcnorm(n,mix=.85)
  o3 <- lv1 + noise*rcnorm(n,mix=.85)

```

```

lv2 <- scale(lv1 + noise*rcnorm(n,mix=.85))
o4 <- lv2 + noise*rcnorm(n,mix=.85)
o5 <- lv2 + noise*rcnorm(n,mix=.85)
o6 <- lv2 + noise*rcnorm(n,mix=.85)
ifelse(effect,
        lv3 <- scale(lv1/2 + lv2/2 + noise*rcnorm(n,mix=.85)),
        lv3 <- scale(lv2 + noise*rcnorm(n,mix=.85)))
o7 <- lv3 + noise*rcnorm(n,mix=.85)
o8 <- lv3 + noise*rcnorm(n,mix=.85)
o9 <- lv3 + noise*rcnorm(n,mix=.85)
x <- cbind(o1,o2,o3,o4,o5,o6,o7,o8,o9)
colnames(x) <- paste0("o",1:9)
return(x)
}

```

These are the two models compared:

```

# These defined outside of loop
model1 <- '
  # measurement model
  lv1 =~ o1 + o2 + o3
  lv2 =~ o4 + o5 + o6
  lv3 =~ o7 + o8 + o9
  # regressions
  lv2 ~ lv1
  lv3 ~ lv1 + lv2
  # residual correlation
  ,

model2 <- '
  # measurement model
  lv1 =~ o1 + o2 + o3
  lv2 =~ o4 + o5 + o6
  lv3 =~ o7 + o8 + o9
  # regressions
  lv2 ~ lv1
  lv3 ~ lv2
  # residual correlation
  ,

```

This shows the fitting procedure for the models:

```
semf <- function(model,corr,n){
  fit <- tryCatch(sem(model=model,sample.cov=corr,sample.nobs=n,
    optim.method="BFGS",optim.force.converged=TRUE),
    error = function(e){})
  return(fit)
}
av <- function(fit1,fit2){
  x <- tryCatch(anova(fit1,fit2)$Pr[2],error = function(e){})
  if(is.null(x)) x <- NA
  return(x)
}
```

This is the simulation when the effect is present:

```
k <- 10000 # **
set.seed(8477)
semvalsET <- matrix(ncol=7,nrow=k)
semvalsET[,1] <- 1:k
for (i in 1:k){
  egchif <- makesemu()
  egnorm <- makesemn()
  egcnorm <- makesemcn()
  cmu1 <- cor(egchif)
  cmu2 <- ncorrmat(egchif)
  cmn1 <- cor(egnorm)
  cmn2 <- ncorrmat(egnorm)
  cmcn1 <- cor(egcnorm)
  cmcn2 <- ncorrmat(egcnorm)

  fitup1 <- semf(model1,cmu1,nrow(egchif))
  fitup2 <- semf(model2,cmu1,nrow(egchif))
  semvalsET[i,2] <- av(fitup1,fitup2)

  fitua1 <- semf(model1,cmu2,nrow(egchif))
  fitua2 <- semf(model2,cmu2,nrow(egchif))
  semvalsET[i,3] <- av(fitua1,fitua2)

  fitnp1 <- semf(model1,cmn1,nrow(egnorm))
  fitnp2 <- semf(model2,cmn1,nrow(egnorm))
  semvalsET[i,4] <- av(fitnp1,fitnp2)

  fitna1 <- semf(model1,cmn2,nrow(egnorm))
  fitna2 <- semf(model2,cmn2,nrow(egnorm))
```

```

semvalsET[i,5] <- av(fitna1,fitna2)

fitcnp1 <- semf(model1,cmcn1,nrow(egcnorm))
fitcnp2 <- semf(model2,cmcn1,nrow(egcnorm))
semvalsET[i,6] <- av(fitcnp1,fitcnp2)

fitcna1 <- semf(model1,cmcn2,nrow(egcnorm))
fitcna2 <- semf(model2,cmcn2,nrow(egcnorm))
semvalsET[i,7] <- av(fitcna1,fitcna2)
}

```

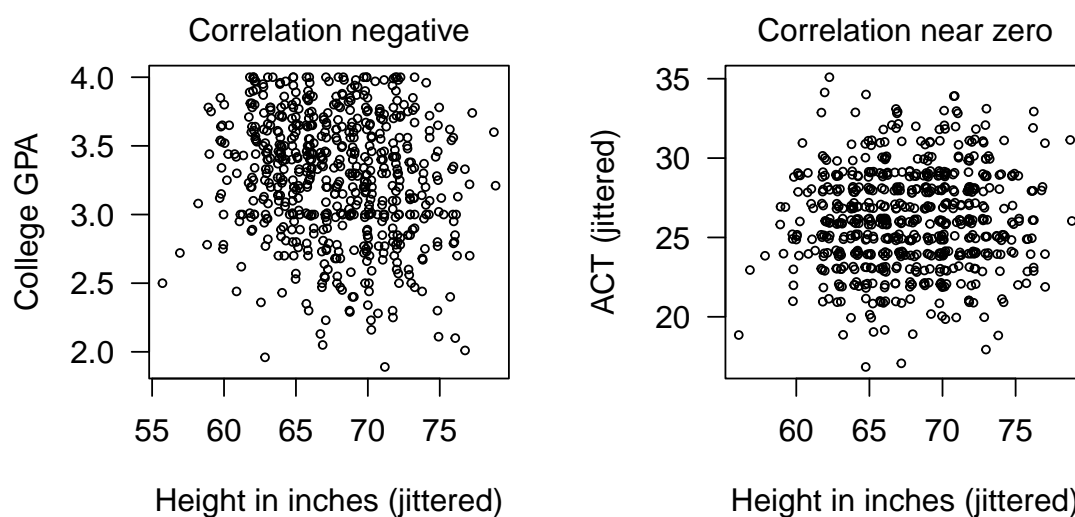
## Appendix 2: Examples using normrank in R

### From Simulation 2: Finding the normrank correlation

The proposed alternative is implemented in three steps: 1) decide how to treat missing values, 2) transform the variables, and 3) run Pearson's correlation. Particularly for procedures like structural equation modeling that use correlation matrices (see below), imputing missing values can be desirable, but it should be done cautiously (?).

#### Figure 7

Scatter plots for height with college GPA and ACT scores for male students in a regression class.



The example dataset used is EDUCATION (?). It is a data set from a regression

course at University of Tennessee, and just the male student data are used here (there are no missing data). The interest is with the association between height and the academic measures: College GPA and ACT scores. Figure 7 shows the scatter plots for these associations. Once the data are loaded, the transformations can be done in R with:

```
newHeight <- qnorm(rank(Height)/(length(Height)+1))
newCGPA <- qnorm((rank(CollegeGPA)-.7)/(length(CollegeGPA) - 2*.7 + 1))
newACT <- qnorm((rank(ACT)-.7)/(length(ACT) - 2*.7 + 1))
```

If you plan to use this transformation often or use it in other functions it can be useful to write a function for it.

```
normrank <- function(x,delta)
  qnorm((rank(x) - delta)/(length(x) - 2 * delta + 1))
```

The final step is conducting the correlation. This can be done with the R functions `cor` or `cor.test`. There is a negative association between height and college GPA, but the association between height and ACT scores is near zero. The `cor.test` function provides more information than `cor`, including *p*-values and confidence intervals.

```
cor.test(newHeight,newCGPA)

##
## Pearson's product-moment correlation
##
## data: newHeight and newCGPA
## t = -3.8639, df = 605, p-value = 0.0001236
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.23190484 -0.07655196
## sample estimates:
## cor
## -0.1551877

cor.test(newHeight,newACT)
```



```
##
## Pearson's product-moment correlation
##
## data: newHeight and newACT
## t = 1.612, df = 605, p-value = 0.1075
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.01425778 0.14422812
## sample estimates:
##          cor
## 0.06539758
```

These steps can be combined:

```
ncorr <- function(x,y,delta=.7){
  if (any(is.na(cbind(x,y))))
    warning("Some missing values. NAs treated as maxima.")
  return(cor(normrank(x,delta=delta),normrank(y,delta=delta)))
}
```

Note that by default the R `rank` function treats missing values as maxima, even higher than  $\infty$ :

```
rank(c(1,NA,Inf,2))
## [1] 1 4 3 2
```

This could lead to unexpected results so it is important to address missing values if this is not appropriate. You can set `na.last = "keep"` in the `rank` functions and `use="pairwise.complete.obs"` in the `cor` function if you want to exclude missing data pairwise.

### EFA in R with the normrank correlation

Consider the fictional data ? created for an SPSS anxiety scale available at [www.discoveringstatistics.com/statistics-hell-p/veritas-advanced-topics/factor-analysis-and-pca/](http://www.discoveringstatistics.com/statistics-hell-p/veritas-advanced-topics/factor-analysis-and-pca/). This is an SPSS file so needs to be read into R using one of the packages that does this, for example **foreign** (?) or **haven** (?). Once the data are read

they are called `spssanx`. Only the first 23 columns are read in; these are the scale's questions. It is convenient to have a function designed to create correlation matrices using the normrank correlation, calling that function (the `normrank` function shown earlier).

```
ncorrmat <- function(x){
  if (any(is.na(x)))
    warning("Some missing values. NAs treated as maxima.")
  return(cor(apply(x,2,normrank,delta=.7)))
}
```

Applying this to `spssanx` results in an normrank correlation matrix:

```
altcor <- ncorrmat(spssanx)
```

That can be used to create a scree plot, using the principle component functions in R (e.g., `prcomp`, `princomp`):

```
screeplot(princomp(covmat=altcor))
```

or directly calling `eigen` or `svd` with the correlation matrix:

```
eigen(altcor)$values
svd(altcor)$d
```

The correlation matrix can also be placed within the `factanal` function for factor analysis. Suppose there are three latent variables. The code would be:

```
factanal(covmat=altcor,factors=3,n.obs=nrow(spssanx))
```

## Using the normrank with SEM in R

As with EFA, a correlation matrix can be created by transforming all the variables and creating a matrix, and then reading this matrix into the multivariate procedure. The functions `cfa`, `sem`, `sam`, and `growth`, are all wrappers for the function `lavaan`. They set

up various parameters consistent with those used by most using confirmatory factor analysis, structural equation modeling, structural after measurement, and latent growth modeling, respectively. Consider the model for the simulation:

```
model1 <- '
  lv1 =~ o1 + o2 + o3
  lv2 =~ o4 + o5 + o6
  lv3 =~ o7 + o8 + o9
  lv2 ~ lv1
  lv3 ~ lv1 + lv2
  '
```

The `sem` function can be used to estimate this model with the normrank correlation matrix by using the `ncorrmatrix` function described earlier. The `sem` function has a slot `sample.cov`. The correlation matrix can be placed in this. In addition, the function needs to be told the number of people that the correlation is based on. Suppose the data are in a standard rectangular data matrix called `semegdata`. The sample size is the number of rows: `nrow(semegdata)`.

```
sem1 <- sem(model1,
            sample.cov= ncorrmatrix(semegdata),
            sample.nobs=nrow(semegdata))
summary(sem1)
```