TETRACHORIC CORRELATIONS AND THE PEARSON PACKAGE

JAN DE LEEUW

ABSTRACT. We derive and review formulas and computations, many of them with a history of more than a hundred years, that can be used to estimate the tetrachoric correlation coefficient and its bias and asymptotic standard error. We also present some new computational methods. The methods are implemented in the R package `pearson`, which will eventually also have polychoric correlations and polychoric multivariate analysis.

Date: Sunday 31st May, 2009 — 22h 11min — Typeset in PALATINO/EULER.

Contents

List of Tables

List of Figures

1. Introduction

Consider the set $\mathcal{P}_{2\times 2}$ of all 2×2, or *four-fold*, probability tables. These tables form a three-dimensional simplex $\mathbb{S}^3$ in four-dimensional space $\mathbb{R}^4$. The four elements of the table are non-negative and add up to one. The interior of $\mathcal{P}_{2\times 2}$, the set of all four-fold tables with all four cells positive, is written as $\mathcal{P}^\circ_{2\times 2}$.

For a table $P \in \mathcal{P}_{2\times 2}$ we use the classical notation

	Y	$\overline{Y}$	
X	p_a	p_b	$p_a + p_b$
$\overline{X}$	p_c	p_d	$p_c + p_d$
	$p_a + p_c$	$p_b + p_d$	1

In this paper, following Pearson [1900], we think of the four-fold table as arising from a discretized bivariate normal distribution. Specifically, we divide the plane into four quadrants using the lines $x = h$ and $y = k$, and we form the four-fold table by integrating the bivariate standard normal density with correlation ρ over the four quadrants.

If $\underline{x}$ and $\underline{y}$ are standard normal random variables [1] with correlation ρ then the four-fold table $P(h, k, \rho)$ is

	Y	$\overline{Y}$	
X	$\mathbf{prob}(\underline{x} < h \wedge \underline{y} < k)$	$\mathbf{prob}(\underline{x} < h \wedge \underline{y} > k)$	$\mathbf{prob}(\underline{x} < h)$
$\overline{X}$	$\mathbf{prob}(\underline{x} > h \wedge \underline{y} < k)$	$\mathbf{prob}(\underline{x} > h \wedge \underline{y} > k)$	$\mathbf{prob}(\underline{x} > h)$
	$\mathbf{prob}(\underline{y} < k)$	$\mathbf{prob}(\underline{y} > k)$	1

When $\rho \in \mathbb{U}$, the open interval $(-1, +1)$, we can write $P(h, k, \rho)$ as

	Y	$\overline{Y}$	
X	$\int_{-\infty}^{h}\int_{-\infty}^{k}\phi(x,y,\rho)dxdy$	$\int_{-\infty}^{h}\int_{k}^{+\infty}\phi(x,y,\rho)dxdy$	$\int_{-\infty}^{h}\phi(x)dx$
$\overline{X}$	$\int_{h}^{+\infty}\int_{-\infty}^{k}\phi(x,y,\rho)dxdy$	$\int_{h}^{+\infty}\int_{k}^{+\infty}\phi(x,y,\rho)dxdy$	$\int_{h}^{+\infty}\phi(x)dx$
	$\int_{-\infty}^{k}\phi(y)dy$	$\int_{k}^{+\infty}\phi(y)dy$	1

[1] Random variables are underlined, cf. Hemelrijk [1966].

where

$$\phi(x,y,\rho) \stackrel{\Delta}{=} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2}\frac{x^2+y^2-2\rho xy}{1-\rho^2}\right\}, \tag{1a}$$

and

$$\phi(x) \stackrel{\Delta}{=} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\}. \tag{1b}$$

Pearson [1900] showed that the map $P(h,k,\rho)$ of $\mathbb{R} \otimes \mathbb{R} \otimes \mathbb{U}$ into $\mathcal{P}^{\circ}_{2\times 2}$ is one-to-one, and consequently invertible. For each $P \in \mathcal{P}^{\circ}_{2\times 2}$ we can find $-\infty < h,k < +\infty$ and $-1 < \rho < +1$ such that $P(h,k,\rho) = P$. The result can be extended to include the boundaries of the regions, and we will do this below. Describing $\mathcal{P}_{2\times 2}$ using the coordinates (h,k,ρ) is known as the *tetrachoric parametrization* [2]. Tetrachoric theory was recently revitalized, and made both more rigorous and more complete, in a very interesting dissertation by Ekström [2009].

2. Bivariate Normal Probabilities

2.1. **Basics.** Define

$$\Phi(h,k,\rho) \stackrel{\Delta}{=} \mathbf{prob}(\underline{x} < h \wedge \underline{y} < k). \tag{2}$$

If $-1 < \rho < +1$

$$\Phi(h,k,\rho) = \int_{-\infty}^{h}\int_{-\infty}^{k} \phi(x,y,\rho)\,dy\,dx. \tag{3}$$

For $\rho = \pm 1$ we go back to the definition (2).

$$\Phi(h,k,1) = \mathbf{prob}(\underline{x} < k \wedge \underline{x} < h) = \min(\Phi(h),\Phi(k)), \tag{4a}$$

and

$$\Phi(h,k,-1) = \mathbf{prob}(\underline{x} < k \wedge -\underline{x} < h) = \max(0,\Phi(h)+\Phi(k)-1), \tag{4b}$$

where

$$\Phi(z) \stackrel{\Delta}{=} \int_{-\infty}^{z} \phi(z)\,dz. \tag{5}$$

[2] It is tempting to go into the metaphysics of tetrachoric correlation, which is at the basis of the famous Pearson-Yule debate. I will resist the temptation, and just refer to MacKenzie [1978, 1981].

We use the notion of conditional densities and distribution to derive some simple identities, useful for subsequent computations. Define

$$\phi(x|y,\rho) \triangleq \frac{1}{\sqrt{1-\rho^2}}\phi\left(\frac{x-\rho y}{\sqrt{1-\rho^2}}\right), \tag{6a}$$

$$\Phi(x|y,\rho) \triangleq \Phi\left(\frac{x-\rho y}{\sqrt{1-\rho^2}}\right). \tag{6b}$$

Then

$$\phi(h,k,\rho) = \phi(h)\phi(k|h,\rho) = \phi(k)\phi(h|k,\rho), \tag{7a}$$

and

$$\Phi(h,k,\rho) = \int_{-\infty}^{h} \phi(x)\Phi(k|x,\rho)dx = \int_{-\infty}^{k} \phi(y)\Phi(h|y,\rho)dy. \tag{7b}$$

2.2. **Partial Derivatives.** For both computational and statistical purposes we need the partial derivatives [3] of Φ.

The first two partials are simple.

$$\mathcal{D}_1\Phi(h,k,\rho) = \phi(h)\Phi(k|h,\rho), \tag{8a}$$

$$\mathcal{D}_2\Phi(h,k,\rho) = \phi(k)\Phi(h|k,\rho). \tag{8b}$$

The third is more complicated. It can be proved by directly by computation, or by differentiating the tetrachoric series expansion of $\Phi(h,k,\rho)$. But it is derived most easily from the identity $\mathcal{D}_3\phi(h,k,\rho) = \mathcal{D}_{12}\phi(h,k,\rho)$ [Plackett, 1954]. Integrate both sides over both h and k and then interchange the order of integration and differentiation on the left-hand side. This gives

$$\mathcal{D}_3\Phi(h,k,\rho) = \phi(h,k,\rho). \tag{8c}$$

Equation (8c) is an essential part of tetrachoric theory. It shows that for all values of h and k the function $\Phi(h,k,\bullet)$ [4] is strictly increasing in ρ.

[3] We use functional notation for derivatives for reasons given, for example, in Spivak [1965, p. 44-45]. For a function f of a single variable the derivative is $\mathcal{D}f$ and the second derivative is $\mathcal{D}(\mathcal{D}f)$, which we simply write as $\mathcal{D}\mathcal{D}f$ or $\mathcal{D}^{(2)}f$. For a function of two variables the partials are $\mathcal{D}_1 f$ and $\mathcal{D}_2 f$. We abbreviate $\mathcal{D}_1(\mathcal{D}_2 f)$ to $\mathcal{D}_{12}f$. $\mathcal{D}f$ is the gradient (the vector of partial derivatives) and $\mathcal{D}\mathcal{D}f$ is the hessian (the matrix of second partials).

[4] To be precise: the function $\Phi(h,k,\bullet)$ is a function of ρ only, with $\Phi(h,k,\bullet)(\rho) = \Phi(h,k,\rho)$.

2.3. **Second Partials.** The second partials with respect to the integration bounds are

$$\mathcal{D}_{11}\Phi(h,k,\rho) = -\rho\phi(h,k,\rho) - h\phi(h)\Phi(k|h,\rho), \tag{8d}$$

$$\mathcal{D}_{12}\Phi(h,k,\rho) = \quad \phi(h,k,\rho), \tag{8e}$$

$$\mathcal{D}_{22}\Phi(h,k,\rho) = -\rho\phi(h,k,\rho) - h\phi(h)\Phi(k|h,\rho) \tag{8f}$$

The mixed second partials with respect to an integration bound and the correlation are

$$\mathcal{D}_{13}\Phi(h,k,\rho) = \phi(h,k,\rho)\left[\frac{\rho k - h}{1-\rho^2}\right], \tag{8g}$$

$$\mathcal{D}_{23}\Phi(h,k,\rho) = \phi(h,k,\rho)\left[\frac{\rho h - k}{1-\rho^2}\right], \tag{8h}$$

Finally

$$\mathcal{D}_{33}\Phi(h,k,\rho) = -\phi(h,k,\rho)\left[\frac{\rho^3 - hk\rho^2 + (h^2+k^2-1)\rho - hk}{(1-\rho^2)^2}\right]. \tag{8i}$$

Equation (8i) was also given by Iyengar and Tong [1987]. Like (8c) it is of interest to tetrachoric theory, because the sign of $\mathcal{D}_{33}\Phi(h,k,\rho)$ shows where $\Phi(h,k,\bullet)$ is an increasing convex or concave function of ρ. From (8i) the sign of $\mathcal{D}_{33}\Phi(h,k,\rho)$ is the sign of $-\rho^3 + hk\rho^2 - (h^2+k^2-1)\rho + hk$. For fixed h and k this defines a cubic in ρ. Since the cubic can have at most three real roots, this means the interval $[-1,+1]$ is partitioned by the roots into at most four subintervals. Throughout each of these subintervals $\Phi(h,k,\bullet)$ is either convex or concave. We study this "convexity cubic" in more detail in the next section.

2.4. **The Convexity Cubic.** Suppose h and k are real numbers. Assume for the time being $h \neq k$ and $h \neq -k$. Consider the cubic f_{hk} defined by

$$f_{hk}(\rho) = -\rho^3 + hk\rho^2 - (h^2+k^2-1)\rho + hk. \tag{9}$$

We are interested in the real solutions of $f_{hk}(\rho) = 0$. Because f is a cubic

- there is either one real root and two conjugate complex ones or,
- there are three real roots (which can be multiple roots).

Since $f_{hk}(-1) = (h+k)^2 > 0$ and $f_{hk}(+1) = -(h-k)^2 < 0$ there is at least one real root between -1 and $+1$. Suppose there are three real roots, call them α, β and γ. By Viera's rule

$$(\alpha-1)(\beta-1)(\gamma-1) = -(h+k)^2 < 0,$$
$$(\alpha+1)(\beta+1)(\gamma+1) = +(h-k)^2 > 0.$$

If $-1 < \alpha < +1$ it follows that either $\beta, \gamma < -1$ or $\beta, \gamma > +1$ or $-1 < \beta, \gamma < +1$. If $\beta, \gamma < -1$ then f_{hk} must have two critical values less than -1, and Df_{hk} must have two roots less than -1 as well.

The discriminant of f_{hk} is

$$\Delta_0(h,k) \triangleq -4h^4k^4 + h^2k^2(h^2+k^2-1)^2 - 4(h^2+k^2-1)^3 + \\ + 18(h^2+k^2-1)h^2k^2 - 27h^2k^2.$$

If $\Delta_0(h,k) > 0$ there are three distinct real roots, if $\Delta_0(h,k) < 0$ there is one reat real root and two conjugate complex ones. The derivative of f_{hk} is the quadratic

$$\mathcal{D}f_{hk}(\rho) = -3\rho^2 + 2hk\rho - (h^2+k^2-1).$$

Note that $\mathcal{D}f_{hk}(1) = -(h-k)^2 - 2 < 0$ and $\mathcal{D}f_{hk}(-1) = -(h+k)^2 - 2 < 0$. The discriminant of $\mathcal{D}f_{hk}$ is

$$\Delta_1(h,k) \triangleq 4h^2k^2 - 12(h^2+k^2-1).$$

If the discriminant $\Delta_1(h,k)$ is negative, then f_{hk} has no critical values and is strictly decreasing everywhere, with a single root between -1 and $+1$. If there are three distinct real roots there must be two distinct critical values. Thus $\Delta_0(h,k) > 0$ implies $\Delta_1(h,k) > 0$.

In Figure 1 we have made contour plots of $\Delta_0(h,k)$ and $\Delta_1(h,k)$. R code for drawing this figure is in Appendix C.4. The red lines are the points (h,k) for which $\Delta_0(h,k) = 0$, the blue lines have $\Delta_1(h,k) = 0$.

Insert Figure 1 *about here*

- The green region in the plot has both $\Delta_0(h,k) > 0$ and $\Delta_1(h,k) > 0$. The function f_{hk} has three distinct real roots and two critical points.
- The magenta region has $\Delta_0(h,k) < 0$ and $\Delta_1(h,k) > 0$. There are two critical points, but only a single real root (between -1 and $+1$).
- The orange region has $\Delta_0(h,k) < 0$ and $\Delta_1(h,k) < 0$. There is only one real root (between -1 and $+1$) and f_{hk} is everywhere decreasing.

In the magenta and orange regions the function $\Phi(h,k,\rho)$ is convex in $(-1,\alpha)$ and concave in $(\alpha,+1)$.

In the green region there are three real roots $\alpha < \beta < \gamma$, none of them equal to ± 1. If all of them are in $(-1,+1)$, then $\Phi(h,k,\bullet)$ is convex in $(-1,\alpha)$, concave in (α,β), convex in (β,γ), and concave again in $(\gamma,+1)$. If only one root is in $(-1,+1)$ then it is either the smallest root α or the largest root γ. In either case $\Phi(h,k,\bullet)$ is convex in the interval between -1 and the root, and concave in the interval between the root and $+1$.

If $\Delta_0(h,k) = 0$ the cubic has three real roots, which are not distinct. There is either one root with multiplicity three, or two roots with multiplicities one and two. If there is one root with multiplicity three, then it is between -1 and $+1$. Again $\Phi(h,k,\rho)$ is convex in the interval between -1 and the root, and concave in the interval between the root and $+1$. If there is a root with multiplicity one and one with multiplicity two, and only one of them is in $(-1,+1)$, then that is the root with multiplicity one, and $\Phi(h,k,\rho)$ is convex in the interval between -1 and the root, and concave in the interval between the root and $+1$. If both roots are in $(-1,+1)$, then $\Phi(h,k,\rho)$ is convex between -1 and the smallest root and concave between the largest root and $+1$. If the smallest root has multiplicity two, the function is convex between the roots, if the largest root has multiplicity two, then the function is concave between the roots.

If $h = k$ there are some simplifications. We can write

$$f_{hh}(\rho) = -(\rho-1)(\rho-\rho_1(h))(\rho-\rho_2(h)),$$

where

$$\rho_{1,2}(h) = \frac{(h^2-1) \pm \sqrt{(h^2-1)^2-4h^2}}{2}.$$

Define the discriminant

$$\Delta(h) \stackrel{\Delta}{=} (h^2-1)^2 - 4h^2 = (h^2+2h-1)(h^2-2h-1)$$

Then

$$\begin{array}{rcccl} -\infty & < h < & -1-\sqrt{2} & \Rightarrow & \Delta(h) > 0 \\ -1-\sqrt{2} & < h < & +1-\sqrt{2} & \Rightarrow & \Delta(h) < 0 \\ +1-\sqrt{2} & < h < & -1+\sqrt{2} & \Rightarrow & \Delta(h) > 0 \\ -1+\sqrt{2} & < h < & +1+\sqrt{2} & \Rightarrow & \Delta(h) < 0 \\ +1+\sqrt{2} & < h < & +\infty & \Rightarrow & \Delta(h) > 0 \end{array}$$

If $\Delta(h) < 0$ then there is a single real root equal to one, and $\Phi(h, h, \bullet)$ is convex on $[-1, +1]$.

If $h = -k$ we have a root at -1 and

$$f_{hh}(\rho) = -(\rho + 1)(\rho^2 + (h^2 - 1)\rho + h^2)$$

The discriminant of the quadratic residual is the same as for $h = k$, and thus there are two more real roots for the same values of h. The two remaining roots are

$$\rho_{1,2} = \frac{-(h^2 - 1) \pm \sqrt{(h^2 - 1)^2 - 4h^2}}{2}.$$

In Figure 2 we have plotted $\Phi(h, k, \rho)$ as a function of ρ for four values of (h, k). Although the function is always smooth and strictly increasing, its shape varies.

Insert Figure 2 *about here*

The vertical blue lines indicate the zeroes of the convexity cubic, which is plotted seperately for the same (h, k) combination in Figure 3.

Insert Figure 3 *about here*

2.5. **Computation.**

2.5.1. *bivnorm.* The file `bivorm.R` in Appendix C.2 has some additional R code to compute bivariate normal probabilities. The function `bivnorm` calls the `f77` routine from `tvpack`, described below. This is the same code used in the bivariate normal computations in Genz et al. [2009a].

`bivnormTetra` takes a fixed number of terms from the tetrachoric series (**??**). It is mainly included for didactic and comparison purposes, since the `tvpack` routine is both faster and more precise. The function uses a maximum of `itmax` terms, but stops if `nsuc` terms are less that `eps` in absolute value.

The function `bivnormTable` takes as its arguments two vectors `x` and `y`, and a correlation `r`, and computes a bivariate normal table with discretizations at `x` and `y`.

`PhiRhoPlotter(h,k)` plots $\Phi(h, k, \rho)$ as a function of ρ, and optionally puts in the zeroes of the convexity cubic. It was used to make the plots in Figure 2 .

`dbNorm(x,y,rho)` gives the density of the bivariate normal.

2.5.2. *tvpack.* The code in Appendix C.3 provides an R interface to the `f77` code of `tvpack`, which implements one-dimensional, two-dimensional and three-dimensional rectangular normal and Student `t` probabilities [Genz, 2004]. The code for the `f77` subroutines is on Alan Genz website at the

http://www.math.wsu.edu/faculty/genz/software/fort77/tvpack.f

Because `tvpack` provides `f77` functions returning doubles, some code was added to wrap the functions in subroutines, such that the `.Fortran` interface from R could be used.

The R calling functions are `bivNorm, bivT, trNorm, triT, gNorm` and `gNorm`. Although the trivariate functions and the `t` functions are not used in this paper, we have included them for completeness.

2.5.3. *cubic.*

3. Tetrachoric Correlation

Theorem 3.1. *If*

$$\max(0, \Phi(k) + \Phi(h) - 1) < p < \min(\Phi(h), \Phi(k))$$

then $\Phi(h, k, \rho) = p$ *has a unique solution* $-1 < \rho(h, k, p) < +1$.

Proof. From (8c), (4b), and (4a) we see that $\Phi(h, k, \rho)$ is a differentiable and strictly increasing function from the open interval $(-1, +1)$ to the open interval $(\max(0, \Phi(k) + \Phi(h) - 1), \min(\Phi(h), \Phi(k)))$. Consequently it has a differentiable and strictly increasing inverse. □

In order to invert the tetrachoric map, i.e. to compute the parameters (h, k, ρ) from a given table P, we must solve the equations

$$\Phi(h) = p_a + p_b, \tag{10a}$$

$$\Phi(k) = p_a + p_c, \tag{10b}$$

$$\Phi(h, k, \rho) = p_a, \tag{10c}$$

for the three unknowns. Parameters h and k are computed by using the quantile functions, i.e. $h(P) = \Phi^{-1}(p_a + p_b)$ and $k(P) = \Phi^{-1}(p_a + p_c)$. Then, using these $h(P)$ and $k(P)$, we solve (10c), which always has a solution because of theorem 3.1.

The bounds from theorem 3.1 become $p_a - p_d \leqslant p_a \leqslant p_a + \min(p_b, p_c)$. If $p_a = p_a + \min(p_b, p_c)$, i.e. $p_b = 0$ or $p_c = 0$, we can choose $\rho(P) = +1$. If $p_a = \max(0, p_a - p_d)$, i.e. $p_a = 0$ or $p_d = 0$, then we can choose $\rho(P) = -1$. We show more precisely how to deal with zeroes in Table 1, given in Appendix A.

Insert Table 1 *about here*

The tetrachoric correlation is not uniquely determined (identified) if and only if a row or a column of the table is zero. In that case all values in the closed interval $[-1, +1]$ are solutions of (10). Note that $h(P)$ and $k(P)$ are always determined uniquely, although they can be infinite. We also see that (10) has a unique solution $-1 < \rho(P) < +1$ if and only if all four elements of the table are positive.

3.1. **Computation.** Computing a tetrachoric correlation coefficient means, from the computational point of view, solve the equation

$$\Phi(h, k, \rho) = p \tag{11}$$

for ρ. Computing h and k is trivial these days, because all computing environments and numerical libraries have high precision quantile functions.

We can first eliminate cases in which there are zero cells, using table 1, and assume without loss of generality that there is a unique root in the open interval $(-1, +1)$. Alternatively, it has been suggested to add $\frac{1}{2}$ to all zero cells, and adjust the non-zero cells to keep the same marginals.

It should be emphasized that for a four-fold table we can actually solve (11) for any of the four p values. If we use p_a or p_d we find ρ, if we use p_b or p_c we compute $-\rho$. This maybe helpful to deal with rounding errors if some proportions are very small or very close to one.

Traditionally, Pearson's tetrachoric series was used to solve (11). The series was truncated after a finite number of terms, and then the resulting polynomial equation was solved for ρ. This is not necessarily the best way to proceed, because for $|\rho|$ close to one the series converges very slowly. Because the derivative of $\Phi(h, k, \rho)$ is so easy to compute, it makes sense to consider Newton's method or one of its many variations [Digvi, 1979]. The package in R that can be used to compute tetrachoric correlations is `polycor` by Fox [2009], which actually computes polychoric correlations using the maximum lkelihood method proposed by Olsson [1979]. Using general polychoric techniques may not be the best way to compute tetrachoric correlations.

Newton's method, unmodified, is

$$\rho^{(k+1)} = \rho^{(k)} - \frac{\Phi(h, k, rho^{(k)}) - p}{\phi(h, k, rho^{(k)})}.$$

To use this iteration we need a good routine to compute bivariate normal probabilities. The ones currently available in R are based on the R implementation of Owen's T function in `sn` by Azzalini [2009], or on `mvtnorm` by Genz et al. [2009b] that actually provides general multivariate normal probabilities over rectangles. We have written code using the tetrachoric series, and a R interface to the compiled code of Genz [2004]. See Appendix C. Some comparisons, both in terms of speed and precision, are needed.

For higher order methods, such as Halley's method, we may need the second partial $\mathcal{D}_{33}\Phi(h, k, \rho)$. The cubic in the numerator is analyzed in detail in Section 2.4. Computing the roots of the cubic allows us to describe an interval in which $\Phi(x, y, \rho)$ has a root and is either convex increasing or concave increasing. This additional knowledge allows us to use powerful versions of Newton's method as well.

4. Statistics

4.1. Standard Errors. Assume all four cells are positive. We can differentiate the three equations in (10) with respect to all seven variables

	h	k	ρ	p_a	p_b	p_c	p_d
Eq (10a)	$\phi(h)$	0	0	–1	–1	0	0
Eq (10b)	0	$\phi(k)$	0	–1	0	–1	0
Eq (10c)	$\phi(h)\Phi(k\vert h,\rho)$	$\phi(k)\Phi(h\vert k,\rho)$	$\phi(h,k,\rho)$	–1	0	0	0

If A is the leading 3×3 lower triangular part of the table, and B is the remaining 3×4 table, then the implicit function theorem tell us the derivatives of $(h(P), k(P), \rho(P))$ with respect to P are given by $-A^{-1}B$. The inverse of A is

$$\begin{bmatrix} \frac{1}{\phi(h)} & 0 & 0 \\ 0 & \frac{1}{\phi(k)} & 0 \\ -\frac{\Phi(k|h,\rho)}{\phi(h,k,\rho)} & -\frac{\Phi(h|k,\rho)}{\phi(h,k,\rho)} & \frac{1}{\phi(h,k,\rho)} \end{bmatrix}$$

and thus the derivatives are

	p_a	p_b	p_c	p_d
h	$\frac{1}{\phi(h)}$	$\frac{1}{\phi(h)}$	0	0
k	$\frac{1}{\phi(k)}$	0	$\frac{1}{\phi(k)}$	0
ρ	$\frac{1-\Phi(k\vert h,\rho)-\Phi(h\vert k,\rho)}{\phi(h,k,\rho)}$	$-\frac{\Phi(k\vert h,\rho)}{\phi(h,k,\rho)}$	$-\frac{\Phi(h\vert k,\rho)}{\phi(h,k,\rho)}$	0

For the Delta method [Mann and Wald, 1943; Tiago De Olivera, 1982] these derivatives are all we need for computation.

4.2. Multinomial Case. Under a multinomial probability model for $\underline{P}$, it follows that the covariance matrix of the asymptotic distribution of $h(\underline{P})$ and $k(\underline{P})$ is

Note that this implies that the correlation between $h(\underline{P})$ and $k(\underline{P})$ in the asymptotic distribution is the population phi-coefficient.

	h	k
h	$\frac{\Phi(h)(1-\Phi(h))}{\phi^2(h)}$	$\frac{\phi(h,k,\rho)-\Phi(h)\Phi(k)}{\phi(h)\phi(k)}$
k	$\frac{\phi(h,k,\rho)-\Phi(h)\Phi(k)}{\phi(h)\phi(k)}$	$\frac{\Phi(k)(1-\Phi(k))}{\phi^2(k)}$

As we know since Pearson [1900], the variance of the asymptotic distribution of $\rho(\underline{P})$ does not look very appealing. Define

$$\mu(h,k,\rho) \stackrel{\Delta}{=} \Phi(h,k,\rho) - \Phi(k|h,\rho)\Phi(h) - \Phi(h|k,\rho)\Phi(k),$$

and

$$\omega(h,k,\rho) \stackrel{\Delta}{=} \{2(1-\Phi(k|h,\rho))(1-\Phi(h|k,\rho)) - 1\}\Phi(h,k,\rho) + \\ + \Phi^2(k|h,\rho)\Phi(h) + \Phi^2(h|k,\rho)\Phi(k).$$

Then

$$\mathbf{AVAR}(\rho(\underline{P})) = \frac{1}{\phi^2(h,k,\rho)}\{\omega(h,k,\rho) - \mu^2(h,k,\rho)\}.$$

It remains to give expressions for the remaining covariances. Let

$$\lambda(h,k,\rho) \stackrel{\Delta}{=} (1-\Phi(h|k,\rho))\Phi(h,k,\rho) - \Phi(k|h,\rho)\Phi(h),$$

$$\eta(h,k,\rho) \stackrel{\Delta}{=} (1-\Phi(k|h,\rho))\Phi(h,k,\rho) - \Phi(h|k,\rho)\Phi(k).$$

Then

$$\mathbf{ACOV}(h(\underline{P}),\rho(\underline{P})) = \frac{1}{\phi(h)\phi(h,k,\rho)}\{\lambda(h,k,\rho) - \Phi(h)\mu(h,k,\rho)\},$$

$$\mathbf{ACOV}(k(\underline{P}),\rho(\underline{P})) = \frac{1}{\phi(k)\phi(h,k,\rho)}\{\eta(h,k,\rho) - \Phi(k)\mu(h,k,\rho)\}.$$

4.3. **Bias.** bias

APPENDIX A. TABLES

p_a	p_b	p_c	p_d	$h(P)$	$k(P)$	$\rho(P)$
+	0	0	0	$+\infty$	$+\infty$	undetermined
0	+	0	0	$+\infty$	$-\infty$	undetermined
0	0	+	0	$-\infty$	$+\infty$	undetermined
0	0	0	+	$-\infty$	$-\infty$	undetermined
+	+	0	0	$+\infty$	$\Phi^{-1}(p_a)$	undetermined
+	0	+	0	$\Phi^{-1}(p_a)$	$+\infty$	undetermined
+	0	0	+	$\Phi^{-1}(p_a)$	$\Phi^{-1}(p_a)$	+1
0	+	+	0	$\Phi^{-1}(p_b)$	$-\Phi^{-1}(p_b)$	–1
0	+	0	+	$\Phi^{-1}(p_b)$	$-\infty$	undetermined
0	0	+	+	$-\infty$	$\Phi^{-1}(p_c)$	undetermined
0	+	+	+	$\Phi^{-1}(p_b)$	$\Phi^{-1}(p_c)$	–1
+	0	+	+	$\Phi^{-1}(p_a)$	$-\Phi^{-1}(p_d)$	+1
+	+	0	+	$-\Phi^{-1}(p_d)$	$\Phi^{-1}(p_a)$	+1
+	+	+	0	$-\Phi^{-1}(p_c)$	$-\Phi^{-1}(p_b)$	–1

TABLE 1. Zeroes in Tetrachoric Correlations

$$\mathcal{D}\mathcal{D}h(P) = \begin{array}{c|cccc} & p_a & p_b & p_c & p_d \\ \hline p_a & -\frac{1}{\phi^3(h)} & -\frac{1}{\phi^3(h)} & 0 & 0 \\ p_b & -\frac{1}{\phi^3(h)} & -\frac{1}{\phi^3(h)} & 0 & 0 \\ p_c & 0 & 0 & 0 & 0 \\ p_d & 0 & 0 & 0 & 0 \end{array}$$

$$\mathcal{D}\mathcal{D}k(P) = \begin{array}{c|cccc} & p_a & p_b & p_c & p_d \\ \hline p_a & -\frac{1}{\phi^3(k)} & 0 & -\frac{1}{\phi^3(k)} & 0 \\ p_b & 0 & 0 & 0 & 0 \\ p_c & -\frac{1}{\phi^3(k)} & 0 & -\frac{1}{\phi^3(k)} & 0 \\ p_d & 0 & 0 & 0 & 0 \end{array}$$

$$\mathcal{D}\mathcal{D}\rho(P) = \begin{array}{c|cccc} & p_a & p_b & p_c & p_d \\ \hline p_a & -\frac{1}{\phi^3(k)} & 0 & -\frac{1}{\phi^3(k)} & 0 \\ p_b & 0 & 0 & 0 & 0 \\ p_c & -\frac{1}{\phi^3(k)} & 0 & -\frac{1}{\phi^3(k)} & 0 \\ p_d & 0 & 0 & 0 & 0 \end{array}$$

APPENDIX B. FIGURES

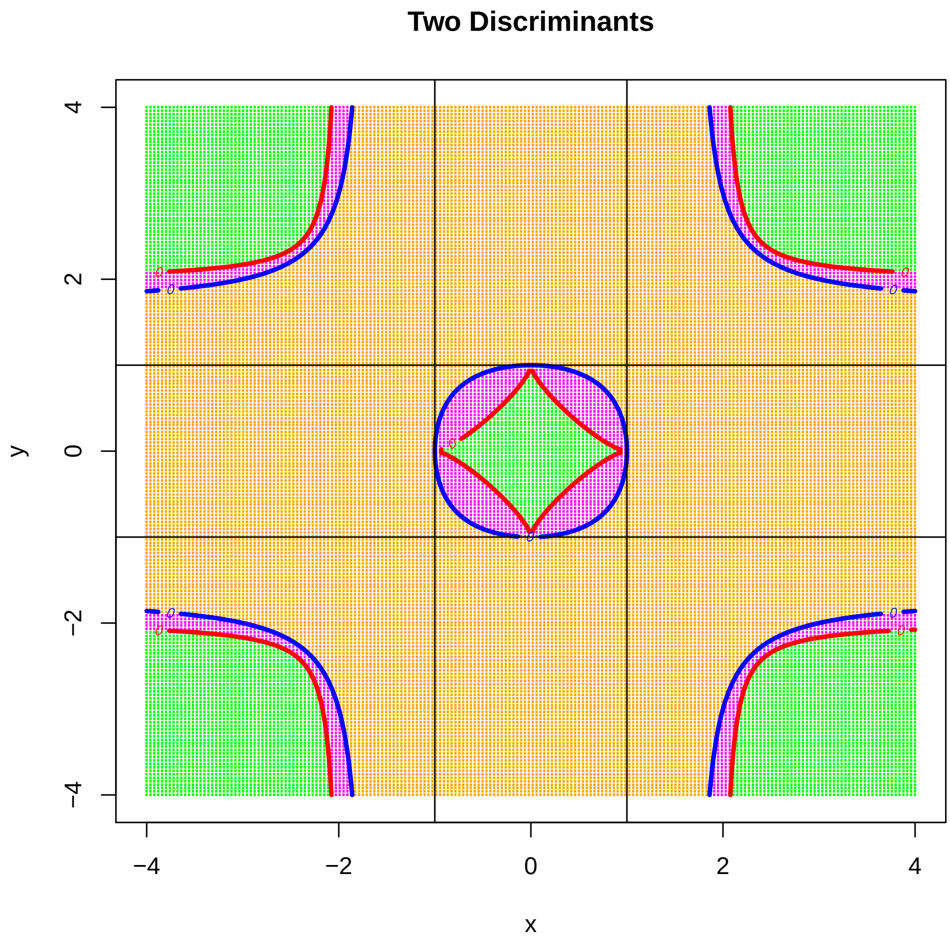

FIGURE 1. Contour Plots Discriminants

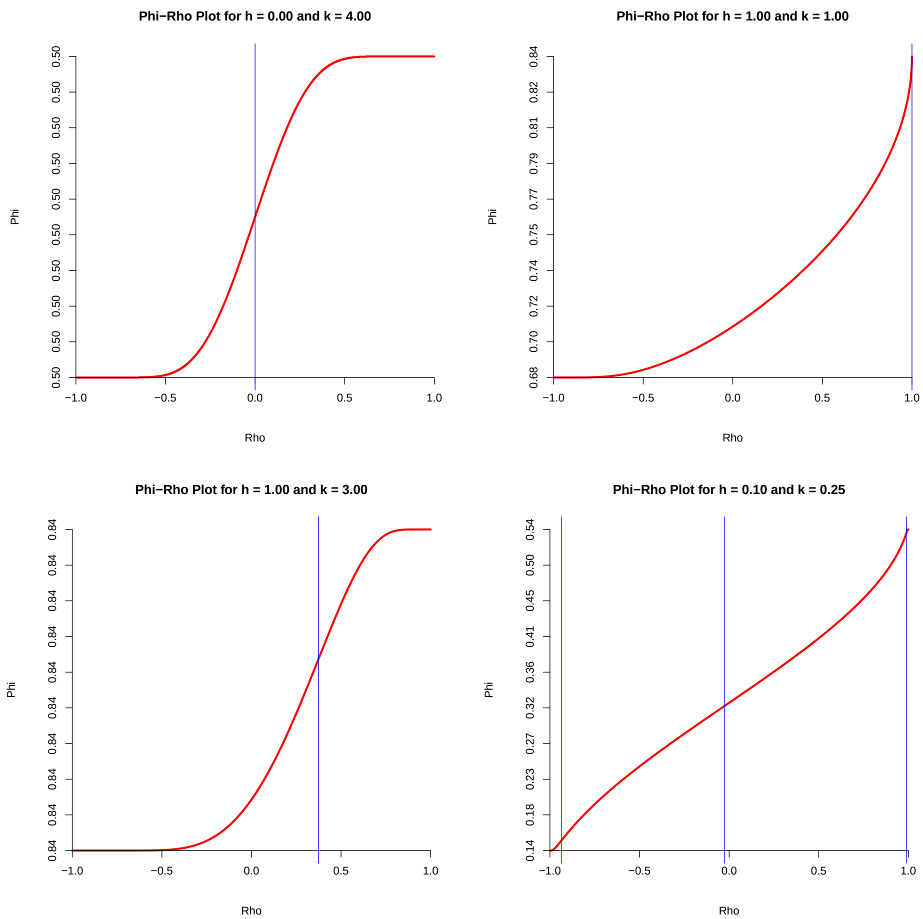

FIGURE 2. PhiRho Plots

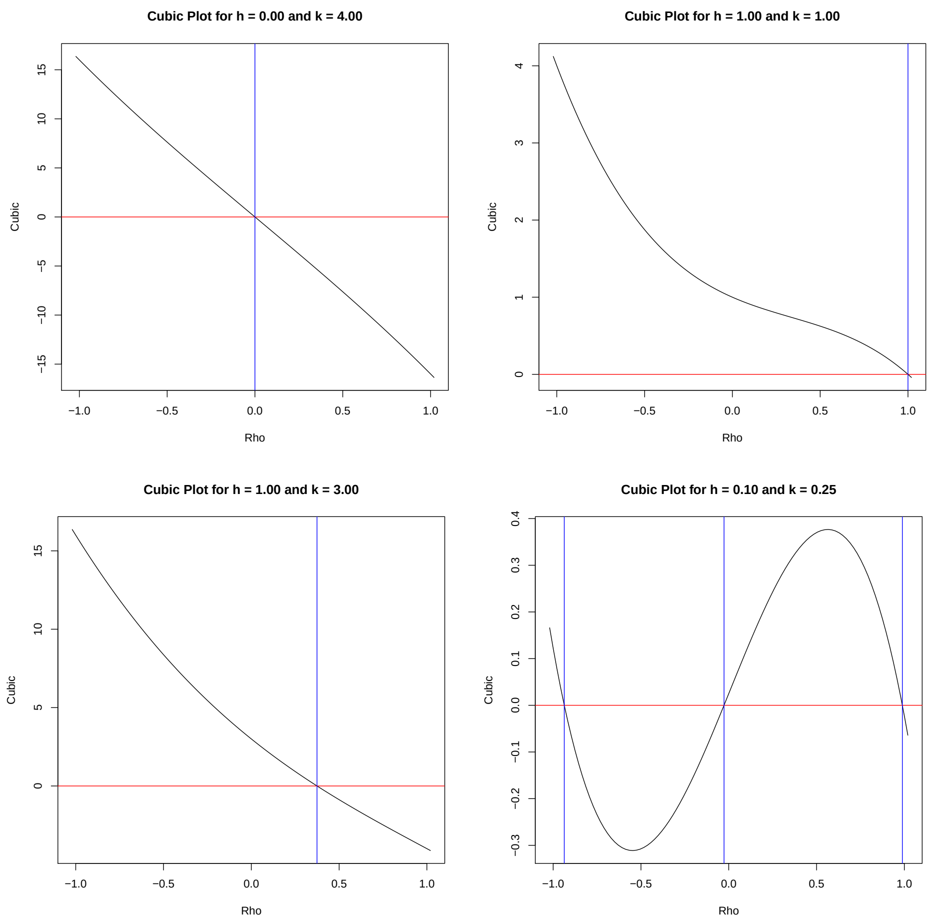

FIGURE 3. Convexity Cubic Plots

APPENDIX C. CODE

C.1. hermite.

C.1.1. *Code in R.*

```
require("polynom")
dyn.load("~/Public/pearson/code/hermite/hermite.so")

hermiteR<-function(x,degree) {
hh<-c(1,x); itel<-1
if (degree > 1)
    repeat {
        uu<-x*hh[2]-itel*hh[1]
        hh[1]<-hh[2]
        hh[2]<-uu
        itel <- itel + 1
        if (itel == degree) break()
        }
else uu<-switch(degree+1,1,x)
return(uu)
}

hermiteF<-function(degree,verbose=FALSE) {
u0<-polynomial(1); u1<-polynomial(c(0,1))
fa<-u0; fb<-u1; itel<-1
switch(degree+1,
    if (verbose) print(fg<-u0),
    if (verbose) print(fg<-u1),
    repeat {
        if (verbose) print(u0)
        if (verbose) print(u1)
        fg<-u1*fb-itel*fa
        if (verbose) print(fg)
        fa<-fb; fb<-fg
        itel<-itel + 1
        if (itel == degree) break()
        }
    )
return(as.function(fg))
```

```
}

tetrachoricR<-function(x,degree) {
if(degree==0) return(pnorm(x))
return(hermiteX(x,degree-1)*dnorm(x)/sqrt(factorial(degree)))
}

tetrachoricF<-function(degree) {
if (degree==0) return(pnorm)
ff<-as.function(hermiteF(degree-1))
return(function(x) ff(x)*dnorm(x)/sqrt(factorial(degree)))
}


hermite<-function(x,degree,norm=0) {
return(.C("hermiteC",x=as.double(x),degree=as.integer(degree),
    norm=as.integer(norm),y=as.double(0))[[4]])
}

tetrachoric<-function(x,degree) {
return(.C("tetrachoricC",x=as.double(x),degree=as.integer(
    degree),y=as.double(0))[[3]])
}
```

C.1.2. *Code in C.*

```
#include <math.h>

void hermiteC(double*, int*, int*, double*);
void tetrachoricC(double* , int*, double*);

void hermiteC(double* x, int* degree, int *norm, double* y) {
double ax = 1.0, bx = (*x), di, dg;
int i = 1, id = (*degree);
switch(id)
    {
    case(0):
        (*y) = ax;
        break;
    case(1):
```

```
        (*y) = bx;
        break;
      default:
        while (i < id) {
            di = (double) i;
            (*y) = (*x) * bx - di * ax;
            ax = bx;
            bx = (*y);
            i++;
            }
    }
if ((*norm) == 1)
    (*y) = (*y) / sqrt(gamma(id + 1.0));
}

void tetrachoricC(double* x, int* degree, double* y) {
double ax = 1.0, bx = (*x), di;
int i = 1, id = (*degree), kd = (*degree) - 1;
double  dg = sqrt(gamma((double) id + 1.0)),
    de = exp(-bx*bx/2.0)/sqrt(2.0 * M_PI),
    df = (erf(bx/M_SQRT2) + 1.0) / 2.0;
switch(kd)
    {
    case(-1):
        (*y) = df;
        break;
    case(0):
        (*y) = de * ax / dg;
        break;
    case(1):
        (*y) = de * bx / dg;
        break;
     default:
        while (i < kd) {
            di = (double) i;
            (*y) = (*x) * bx - di * ax;
            ax = bx;
            bx = (*y);
            i++;
```

```
            }
        (*y) = de * (*y) / dg;
    }
}
```

C.2. bivnorm.

```
dyn.load("~/Public/pearson/code/tvpack/tvpack.so")

bivNormGenz<-function(dh,dk,r) {
out <- .Fortran("BVND_WRAPPER",dh=as.double(-dh),dk=as.double(-dk),r=as.double(r),p=as.double(0))
return(out$p)
}

bivProb<-function(h,k,r) {
out <- .Fortran("BIVPRB_WRAPPER",h=as.double(h),k=as.double(k),r=as.double(r),p=as.double(0))[[4]]
return(out$p)
}

bivNormTetra<-function(x,y,r,nsuc=5,eps=1e-8,itmax=1000,verbose=FALSE) {
sterm<-0; itel<-1; ifac<-1
hxa<-1; hxb<-x; hya<-1; hyb<-y; isuc<-0
sterm<-pnorm(x)*pnorm(y); dxy<-dnorm(x)*dnorm(y)
if (verbose)
   cat("itel ",formatC(0,width=4,format="d"),"term ",formatC(sterm,width=20,digits=12,format="f"),"sum ",formatC(sterm,width=20,digits=12,format="f"),"\n")
repeat {
  term<-(r^itel)*dxy*hxa*hya/ifac
  if (abs(term) < eps) isuc <- isuc +1
  sterm<-sterm+term
  if (verbose)
     cat("itel ",formatC(itel,width=4,format="d"),"term ",formatC(term,width=20,digits=12,format="f"),"sum ",formatC(sterm,width=20,digits=12,format="f"),"\n")
  if ((itel == itmax) || (isuc == nsuc)) return(sterm)
  hx<-x*hxb-itel*hxa; hxa<-hxb; hxb<-hx
```

```
   hy<-y*hyb-itel*hya; hya<-hyb; hyb<-hy
   itel<-itel+1
   ifac<-ifac*itel
   }
}

bivNormTable<-function(x,y,r,eps=1e-6,ntel=10) {
n<-length(x); m<-length(y); f<-matrix(0,n+2,m+2)
nn<-n+1; mm<-m+1; g<-matrix(0,nn,mm); f[n+2,m+2]<-1
for (i in 1:n) {
    for (j in 1:m) {
        f[i+1,j+1]<-bivNormGenz(x[i],y[j],r)
        }
    }
for (j in 1:m) f[n+2,j+1]<-pnorm(y[j])
for (i in 1:n) f[i+1,m+2]<-pnorm(x[i])
for (i in 1:nn) {
    for (j in 1:mm) {
        g[i,j]<-f[i,j]+f[i+1,j+1]-(f[i,j+1]+f[i+1,j])
        }
    }
return(g)
}

PhiRhoPlotter<-function(h,k,npt=1000,cvex=TRUE) {
eps<-1/npt; r<-seq(-1+eps,1-eps,by=eps)
mn<-paste("Phi-Rho Plot for h =",formatC(h,width=4,digits=2,
     format="f"),"and k =",formatC(k,width=4,digits=2,format="f")
     )
ph<-pnorm(h); pk<-pnorm(k); rp<-min(ph,pk); rm<-max(0,ph+pk-1)
p<-sapply(r,function(s) bivNormGenz(-h,-k,s))
plot(c(-1,r,1),c(rm,p,rp),xlab="Rho",ylab="Phi",xlim=c(-1,1),
     ylim=c(rm,rp),main=mn,axes=FALSE,type="l",col="RED",lwd=3)
tcks<-seq(rm,rp,length=10)
labs<-formatC(tcks,width=4,digits=2,flag="#")
axis(2,pos=-1,at=tcks,labels=labs)
axis(1,pos=rm)
if (cvex)
     abline(v=convexCubic(h,k),col="BLUE")
```

```
}

dbNorm<-function(x,y,rho) {
return(exp(-(x^2+y^2-2*rho*x*y)/(2*(1-rho^2)))/(2*pi*sqrt(1-rho^2)))
}
```

C.3. tvpack.

```
dyn.load("tvpack.so")

bivNorm<-function(dh,dk,r) {
out <- .Fortran("BVND_WRAPPER",dh=as.double(dh),dk=as.double(dk),r=as.double(r),p=as.double(0))
return(out$p)
}

bivT<-function(nu,dh,dk,r) {
out <- .Fortran("BVTL_WRAPPER",nu=as.integer(nu),dh=as.double(dh),dk=as.double(dk),r=as.double(r),p=as.double(0))
return(out$p)
}

triNorm<-function(nu,h,r,epsi=1e-8) {
out <- .Fortran("TVTL_WRAPPER",nu=as.integer(nu),h=as.double(h),r=as.double(r),epsi=as.double(epsi),p=as.double(0))
return(out$p)
}

gNorm<-function(z) {
out <- .Fortran("PHID_WRAPPER",z=as.double(z),p=as.double(0))
return(out$p)
}

gT<-function(nu,t) {
out <- .Fortran("STUDNT_WRAPPER",nu=as.integer(nu),t=as.double(t),p=as.double(0))
return(out$p)
}
```

C.4. cubic.

```
require("polynom")

convexCubic<-function(h,k) {
return(cubicR(c(h*k,-(h^2+k^2-1),h*k,-1)))
}

plotCubic<-function(h,k,xmin=-1,xmax=1) {
mn<-paste("Cubic Plot for h =",formatC(h,width=4,digits=2,
    format="f"),"and k =",formatC(k,width=4,digits=2,format="f")
    )
p<-polynomial(c(h*k,-(h^2+k^2-1),h*k,-1))
plot(p,xlim=c(xmin,xmax),main=mn,xlab="Rho",ylab="Cubic")
s<-solve(p); r<-Re(s)
abline(h=0,col="RED")
abline(v=r[r==s],col="BLUE")
}

discriminant<-function(h,k) {
delta_0<--4*h^4*k^4+h^2*k^2*(h^2+k^2-1)^2-4*(h^2+k^2-1)^3+18*(h
    ^2+k^2-1)*h^2*k^2-27*h^2*k^2.
delta_1<-4*h^2*k^2-12*(h^2+k^2-1)
return(c(delta_0,delta_1))
}

contourMe<-function(range=3,grid=200,fill=TRUE) {
x<-y<-seq(-range,range,length=grid)
aa<-outer(x^2,y^2)
cc<-outer(x^2,y^2,"+")-1
zc<--4*aa^2+aa*cc^2-4*cc^3+18*cc*aa-27*aa
zq<-4*aa-12*cc
l<-0
plot(x,y,type="n",main="Two Discriminants")
if (fill)
    for (i in 1:200) for (j in 1:200) {
        if ((zc[i,j] > 0) && (zq[i,j] > 0)) text(x[i],y[j],".",
            col="green")
        if ((zc[i,j] > 0) && (zq[i,j] < 0)) text(x[i],y[j],".",
            col="yellow")
```

```
            if ((zc[i,j] < 0) && (zq[i,j] > 0)) text(x[i],y[j],".",
                col="magenta")
            if ((zc[i,j] < 0) && (zq[i,j] < 0)) text(x[i],y[j],".",
                col="orange")
            }
contour(x,y,zc,levels=1,col="red",lwd=3,vfont = c("sans serif",
     "bold italic"),add=TRUE)
contour(x,y,zq,levels=1,col="blue",lwd=3,vfont = c("sans serif"
    , "bold italic"),add=TRUE)
abline(h=1); abline(h=-1)
abline(v=1); abline(v=-1)
}

cubicR<-function(cf) {
out<-.C("cubic",cf=as.double(cf),rr=as.double(rep(0,3)),num=
    as.integer(0))
return((out$rr)[1:(out$num)])
}
```

C.5. tetra.

```
require("polynom")
dyn.load("/Users/deleeuw/Public/pearson/code/tetra/tetra.so")
source("~/Public/pearson/code/hermite/hermite.R")
source("~/Public/pearson/code/bivnorm/bivnorm.R")

tetraC<-function(table) {
a<-table[1,1]; b<-table[1,2]
c<-table[2,1]; d<-table[2,2]
out<-.Fortran("TETRA",
        as.double(a),
        as.double(b),
        as.double(c),
        as.double(d),
        r=as.double(0),
        sdr=as.double(0),
        sdzero=as.double(0),
        itype=as.integer(0),
        ifault=as.integer(0),DUP=TRUE)
```

```
return(list(r=out$r,sdr=out$sdr,sdzero=out$sdzero,itype=out$itype,ifault=out$ifault))
}

tetraR<-function(table,degree=12) {
ts<-sum(table); ps<-table[1,1]/ts
h<-qnorm((table[1,1]+table[1,2])/ts)
k<-qnorm((table[1,1]+table[2,1])/ts)
v<-table[1,1]*table[2,2]-table[1,2]*table[2,1]
u<-c(v/(dnorm(h)*dnorm(k)),-1); fd<-2
for (d in 1:(degree-1)){
    u<-c(u,-hermite(h,d)*hermite(k,d)/fd)
    fd<-fd*(d+2)
    }
s<-solve(polynomial(u))
s<-Re(s[which(s==Re(s))])
return(list(h=h,k=k,r=s[which(s*s < 1)]))
}

rsolveRA<-function(table) {
ts<-sum(table); ps<-table[1,1]/ts
h<-qnorm((table[1,1]+table[1,2])/ts)
k<-qnorm((table[1,1]+table[2,1])/ts)
b<-function(r) bivNorm(h,k,r)-ps
r<-uniroot(b,c(-1,1))$root
return(list(h=h,k=k,r=r))
}

rsolveRB<-function(table,eps=1e-10,itmax=100,verbose=TRUE) {
ts<-sum(table); ps<-table[1,1]/ts
h<-qnorm((table[1,1]+table[1,2])/ts)
k<-qnorm((table[1,1]+table[2,1])/ts)
f<-function(r) bivNorm(h,k,r)-ps
g<-function(r) dbNorm(h,k,r)
y<-sort(convexCubic(h,k))
print(y)
if (length(y)==1)
    return(newton(y,f,g,eps,itmax,verbose))
if (length(y)==3) {
```

```
    z<-c(f(-1),sapply(y,f),f(1))
print(z)
    k<-which(z>0)[1]
print(k)
    if (k < 4) return(newton(y[1],f,g,eps,itmax,verbose))
        else return(newton(y[3],f,g,eps,itmax,verbose))
    }
}

newtonFD<-function(a,b,f,g,method="F",eps=1e-10,itmax=100,
    verbose=TRUE) {
xold<-a; yold<-b; itel<-0
if (verbose)
    cat("itel",formatC(itel,format="d",width=4)," left",
        formatC(xold,format="f",digits=8,width=15)," right",
        formatC(yold,format="f",digits=8,width=15),"\n")
itel<-1
repeat{
    gold<-g(xold); fold<-f(xold); hold<-f(yold)
    xnew<-xold-fold/gold
    if (method=="F") ynew<-yold-hold/gold
    if (method=="D") ynew<-yold-hold*(yold-xold)/(hold-fold)
    if (method=="S") ynew<-xold
    if (verbose)
        cat("itel",formatC(itel,format="d",width=4)," left",
            formatC(xnew,format="f",digits=8,width=15)," right",
            formatC(ynew,format="f",digits=8,width=15),"\n")
    if ((abs(xnew-ynew) < eps) || (itel == itmax))
        return(list(root=(xnew+ynew)/2,left=xnew,right=ynew,
            itel=itel))
    xold<-xnew; yold<-ynew; itel<-itel+1
    }
}

newton<-function(x,f,g,eps=1e-10,itmax=100,verbose=TRUE) {
xold<-x; itel<-0
if (verbose)
    cat("itel",formatC(itel,format="d",width=4)," root",
        formatC(xold,format="f",digits=8,width=15),"\n")
```

```
itel<-1
repeat{
    gold<-g(xold); fold<-f(xold)
    xnew<-xold-fold/gold
    if (verbose)
        cat("itel",formatC(itel,format="d",width=4)," root",
            formatC(xnew,format="f",digits=8,width=15),"\n")
    if ((abs(xnew-xold) < eps) || (itel == itmax))
        return(list(root=xnew,itel=itel))
    xold<-xnew; itel<-itel+1
    }
}

smallpox<-matrix(c(1562,383,42,94),2,2); smallpox<-smallpox/sum(smallpox)

diphteria<-matrix(c(319,177,143,289),2,2); diphteria<-diphteria/sum(diphteria)

eyecolor<-matrix(c(254,156,136,193),2,2); eyecolor<-eyecolor/sum(eyecolor)

hounds<-matrix(c(1766,842,842,722),2,2); hounds<-hounds/sum(hounds)

horses<-matrix(c(631,147,125,147),2,2); horses<-horses/sum(horses)
```

REFERENCES

A. Azzalini. *R package `sn`: The skew-normal and skew-`t` distributions*. Università di Padova, Italia, 2009. URL `http://azzalini.stat.unipd.it/SN`. R package version 0.4-12.

D.R. Digvi. Calculation of the Tetrachoric Correlation Coefficient. *Psychometrika*, 44:168–172, 1979.

J. Ekström. *Contributions to the Theory of Measures of Association for Ordinal Variables*. PhD thesis, Upssala Universitet, 2009.

J. Fox. *`polycor`: Polychoric and Polyserial Correlations*, 2009. URL `http://CRAN.R-project.org/package=polycor`. R package version 0.7-7.

A. Genz. Numerical Computation of Rectangular Bivariate and Trivariate Normal and `t` Probabilities. *Statistics and Computing*, 14:251–260, 2004. URL `http://www.math.wsu.edu/faculty/genz/software/fort77/tvpack.f`.

A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2009a. URL `http://CRAN.R-project.org/package=mvtnorm`. R package version 0.9-7.

A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, and T. Hothorn. *`mvtnorm`: Multivariate Normal and t Distributions*, 2009b. URL `http://CRAN.R-project.org/package=mvtnorm`. R package version 0.9-6.

J. Hemelrijk. Underlining Random Variables. *Statistica Neerlandica*, 20:1–7, 1966.

U.W. Hochstrasser. Orthogonal Polynomials. In M. Abramowitz and I.A. Stegun, editors, *Handbook of Mathematical Functions*, chapter 22. Dover Publications, New York, N.Y., 1965.

S. Iyengar and Y.L. Tong. Convexity Properties of Elliptically Contoured Distributions with Applications. *Sankhya*, A 51:13–29, 1987.

H.O. Lancaster. The Structure of Bivariate Distributions. *Annals of Mathematical Statistics*, 29:719–736, 1958.

D.A. MacKenzie. Statistical Theory and Social Interest: A Case-Study. *Social Studies of Science*, 8:35–83, 1978.

D.A. MacKenzie. *Statistics in Britain 1865-1930. The Social Construction of Scientific Knowledge.* Edinburgh University Press, Edinburgh, 1981.

H.B. Mann and A. Wald. On Stochastic Limit and Order Relationships. *Annals of Mathematical Statistics*, 14:217–226, 1943.

F.G. Mehler. Über die Entwicklung einer Funktion von beliebig vielen Variabelen nach Laplaceschen Funktionen höherer Ordnung. *Journal für Reine und Angewandte Mathematik*, 66:161–176, 1866.

U. Olsson. Maximum Likelihood Estimation of the Polychoric Correlation Coefficient. *Psychometrika*, 443–460, 1979.

K. Pearson. Mathematical Contributions to the Theory of Evolution VII. On the Correlation of Characters not Quantitatively Measurable. *Philosophical Transactions of the Royal Society*, A 195:1–47, 1900. URL `http://www.jstor.org/page/termsConfirm.jsp?redirectUri=/stable/pdfplus/90764.pdf`.

R.L. Plackett. Reduction Formula for Multivariate Normal Integrals. *Biometrika*, 41:351–360, 1954.

M. Spivak. *Calculus on Manifolds*. W.A. Benjamin, Inc, New York, N.Y., 1965.

J. Tiago De Olivera. The Delta Method for Obtention of Asymptotic Distributions; Applications. *Publications de l'Institute de Statistique de l'Université de Paris*, 27:49–70, 1982.

B. Venables, K. Hornik, and M. Maechler. *`polynom`: A Collection of Functions to Implement a Class for Univariate Polynomial Manipulations*, 2009. URL `http://CRAN.R-project.org/package=polynom`. `R` package version 1.3-5. `S` original by Bill Venables, packages for `R` by Kurt Hornik and Martin Maechler.

M. Zelen and N.C. Severo. Probability Functions. In M. Abramowitz and I.A. Stegun, editors, *Handbook of Mathematical Functions*, chapter 26. Dover Publications, New York, N.Y., 1965.

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90095-1554

E-mail address, Jan de Leeuw: `deleeuw@stat.ucla.edu`

URL, Jan de Leeuw: `http://gifi.stat.ucla.edu`