# snp15.$y$ | Confidence intervals for rank order statistics and their differences

Author: Roger Newson, Imperial College London, UK. Email: r.newson@imperial.ac.uk Date: 16 April 2020.

**Abstract:** Rank order or so–called “non–parametric” methods are in fact based on population parameters, which are zero under the null hypothesis. Two of these parameters are Kendall’s $\tau_a$ and Somers’ $D$, the parameter tested by a Wilcoxon rank–sum test. Confidence limits for these parameters are more informative than $P$–values alone, for three reasons. First, confidence intervals show that a high $P$–value does not prove a null hypothesis. Second, for continuous data, Kendall’s $\tau_a$ can often be used to define robust confidence limits for Pearson’s correlation by Greiner’s relation. Third, we can define confidence limits for differences between two Kendall’s $\tau_a$s or Somers’ $D$s, and these are informative, because a larger Kendall’s $\tau_a$ or Somers’ $D$ cannot be secondary to a smaller one. The program `somersd` calculates confidence intervals for Somers’ $D$ or Kendall’s $\tau_a$, using jackknife variances. There is a choice of transformations, including Fisher’s $z$, Daniels’ arcsine, Greiner’s $\rho$, the $z$–transform of Greiner’s $\rho$, and Harrell’s $c$. A `cluster()` option is available for clustered samples. The `funtype()` option allows the user to estimate between–cluster, within–cluster or von Mises versions of Somers’ $D$ or Kendall’s $\tau_a$. The `wstrata()` option allows the user to estimate stratified versions of Somers’ $D$ or Kendall’s $\tau_a$, which can be used to measure associations between a predictor and an outcome within strata specified by the values of a confounder. The `cenind()` option allows the user to specify left or right censorship indicators for the $X$ and/or $Y$ variables. The estimation results are saved as for a model fit, so that differences can be estimated using `lincom`.

**Keywords:** Somers’ $D$; Kendall’s tau; Harrell’s $c$; ROC area; Gini index; rank correlation; rank–sum test; Wilcoxon test; sign test; confidence intervals; non–parametric methods.

## 1 Syntax

`somersd` [*varlist* ] [*weight* ] [`if` *exp* ] [`in` *range* ] [`, taua tdist transf(`*transformation_name*`)`
`cenind(`*cenind_list*`) cluster(`*varname*`) cfweight(`*expression*`) funtype(`*functional_type*`)`
`wstrata(`*varlist*`) bstrata(`*varlist* `| _n) notree level(`*#*`) cimatrix(`*new_matrix*`)` ]

where *transformation_name* is one of

`iden | z | asin | rho | zrho | c`

and *functional_type* is one of

`wcluster | bcluster | vonmises`

and *cenind_list* is a list of variable names and/or zeros.

`fweights`, `iweights` and `pweights` are allowed; see help for `weight`. They are treated as described in **Interpretation of weights** and **Methods and formulas** below.

`bootstrap`, `by`, `jackknife`, `statsby`, `mi impute`, `svy jackknife`, `svy bootstrap`, `svy brr` and `svy sdr` are allowed; see help for `prefix`.

### 1.1 Description

The `somersd` package computes confidence intervals for a wide range of rank statistics. It includes 3 component modules, each with a `.pdf` manual, as follows:

| *Module* | *Manual* | *Calculates confidence intervals for:* |
|---|---|---|
| `somersd` | `somersd.pdf` | Kendall’s tau-a and Somers’ D |
| `censlope` | `censlope.pdf` | Theil-Sen median (and other percentile) slopes |
| `cendif` | `cendif.pdf` | Hodges-Lehmann median (and other percentile) differences |

The modules `censlope` and `cendif` require the module `somersd` in order to work, and use some of the options of `somersd`.

`somersd` calculates the rank order statistics Somers’ $D$ (corresponding to rank–sum tests) and Kendall’s $\tau_a$, with confidence limits. Somers’ $D$ or Kendall’s $\tau_a$ is calculated for the first variable of *varlist* as a predictor of each of the other variables in *varlist*, with estimates and jackknife variances and confidence intervals output and saved in `e()` as if for the parameters of a model fit. It is possible to use `lincom` to output confidence limits for differences between the population Somers’ $D$ or Kendall’s $\tau_a$ values.

### 1.2 Options

`taua` causes `somersd` to calculate Kendall’s $\tau_a$. If `taua` is absent, then `somersd` calculates Somers’ $D$.

tdist specifies that the estimates are assumed to have a $t$–distribution with $N-1$ degrees of freedom, where $N$ is the number of clusters if cluster() is specified, or the number of observations if cluster() is not specified. If tdist is not specified, then the standardized Somers' $D$ estimates are assumed to be sampled from a standard Normal distribution. Simulation study data suggest that the tdist option should be recommended.

transf(*transformation_name*) specifies that the estimates are to be transformed, defining estimates for the transformed population value. iden (identity or untransformed) is the default. z specifies Fisher's $z$ (the hyperbolic arctangent), asin specifies Daniels' arcsine, rho specifies Greiner's $\rho$ (Pearson correlation estimated using Greiner's relation), zrho specifies the $z$–transform of Greiner's $\rho$, and c specifies Harrell's $c$. If the first variable of *varlist* is a binary indicator of a disease and the other variables are quantitative predictors for that disease, then Harrell's $c$ is the area under the receiver operating characteristic (ROC) curve. somersd recognises the transformation names arctanh and atanh as synonyms for z, arcsin and arsin as synonyms for asin, sinph as a synonym for rho, zsinph as a synonym for zrho, and roc and auroc as synonyms for c. It also recognizes unambiguous abbreviations for transformation names, such as id for iden, or aur for auroc.

cenind(*cenind_list*) specifies a list of left– or right–censorship indicators, corresponding to the variables mentioned in the *varlist.* Each censorship indicator is either a variable name or a zero. If the censorship indicator corresponding to a variable is the name of a second variable, then this second variable is used to indicate the censorship status of the first variable, which is assumed to be left–censored (at or below its stated value) in observations in which the second variable is negative, right–censored (at or above its stated value) in observations in which the second variable is positive, and uncensored (equal to its stated value) in observations in which the second variable is zero. If the censorship indicator corresponding to a variable is a zero, then the variable is assumed to be uncensored. If cenind() is unspecified, then all variables in the *varlist* are assumed to be uncensored. If the list of censorship indicators specified by cenind() is shorter than the list of variables specified in the *varlist*, then the list of censorship indicators is completed with the required number of zeros on the right.

cluster(*varname*) specifies the variable which defines sampling clusters. If cluster() is defined, then the variances and confidence limits are calculated assuming that the data represent a sample of clusters from a population of clusters, rather than a sample of observations from a population of observations.

cfweight(*expression*) specifies an expression giving the cluster frequency weights. These cluster frequency weights must have the same value for all observations in a cluster. If cfweight() and cluster() are both specified, then each cluster in the dataset is assumed to represent a number of identical clusters equal to the cluster frequency weight for that cluster. If cfweight() is specified and cluster() is unspecified, then each observation in the dataset is treated as a cluster, and assumed to represent a number of identical one–observation clusters equal to the cluster frequency weight. For more details on the interpretation of weights, see **Interpretation of weights** below.

funtype(*functional_type*) specifies whether the Somers' $D$ or Kendall's $\tau_a$ functionals estimated are ratios of between–cluster, within–cluster or von Mises functionals. These three functional types are specified by the options funtype(bcluster), funtype(wcluster) or funtype(vonmises), respectively. If funtype() is not specified, then funtype(bcluster) is assumed, and between–cluster functionals are estimated. The within–cluster Somers' $D$ is a generalization of the confidence interval corresponding to the sign test (see [R] signrank). The Gini coefficient is a special case of the clustered von Mises Somers' $D$. For further details, see **Methods and Formulas**.

wstrata(*varlist*) specifies a list of variables whose value combinations are the $W$–strata. If wstrata() is specified, then somersd estimates stratified Somers' $D$ or Kendall's $\tau_a$ parameters, applying only to pairs of observations within the same $W$–stratum. These parameters can be used to measure associations within strata, such as associations between an outcome and an exposure within groups defined by values of a confounder, or by values of a propensity score based on multiple confounders.

bstrata(*varlist* | _n) specifies the $B$–strata. If bstrata() is specified, then somersd estimates Somers' $D$ or Kendall's $\tau_a$ parameters specific to pairs of observations from different $B$–strata. These $B$–strata are either combinations of values of a list of variables (if *varlist* is specified) or the individual observations (if _n is specified). $B$–strata will not often be required. However, if we are estimating the within–cluster Kendall's $\tau_a$ (using the options taua funtype(wcluster)), then the additional option bstrata(_n) will ensure that the within–cluster Kendall's $\tau_a$ can take the whole range of values from $-1$ (in the case of complete discordance within clusters) to $+1$ (in the case of complete concordance within clusters).

notree specifies that somersd does not use the default search tree algorithm based on Newson (2006a), but instead uses a trivial algorithm, which compares every pair of observations and requires much more time with large datasets. This option is rarely used except to compare performance.

`level(`*#*`)` specifies the confidence level, in percent, for confidence intervals of the estimates; see [R] `level`.

`cimatrix(`*new_matrix*`)` specifies an output matrix to be created, containing estimates and confidence limits for the untransformed Somers' $D$, Kendall's $\tau_a$ or Greiner's $\rho$ parameters. If `transf()` is specified, then the confidence limits will be asymmetric and based on symmetric confidence limits for the transformed parameters. This option (like `level()`) may be used in replay mode as well as in non–replay mode.

If a *varlist* is supplied, then all options are allowed. If not, then `somersd` replays the previous `somersd` estimation (if available), and the only options allowed are `level()` and `cimatrix()`.

### 1.3 Interpretation of weights

`somersd` inputs up to two weight expressions, which are the ordinary Stata weights given by the *weight* and the cluster frequency weights given by the `cfweight()` option. Internally, `somersd` defines and uses three distinct sets of weights, which are the cluster frequency weights, the observation frequency weights, and the importance weights.

The cluster frequency weights must be the same for different observations in a cluster, and imply that each cluster in the input dataset represents a number of identical clusters equal to the cluster frequency weight in that cluster. If `cluster()` is not specified, then the individual observations are clusters, and the cluster frequency weight implies that each one–observation cluster represents a number of identical one–observation clusters equal to the cluster frequency weight. The cluster frequency weights are given by `cfweight()` if that option is specified, are set to 1 if `cfweight()` is unspecified and `cluster()` is specified, are equal to the ordinary Stata weights if neither `cluster()` nor `cfweight()` is specified and the ordinary Stata weights are `fweight`s, and are equal to 1 otherwise.

The observation frequency weights are summed over all observations in the input dataset to produce the number of observations reported by `somersd` and returned in the estimation result `e(N)`, and are not used in any other way. They are set by `cfweight()` if that option is specified and the ordinary Stata weights are not `fweight`s, are equal to the ordinary Stata weights if `cfweight()` is unspecified and the ordinary Stata weights are `fweight`s, are equal to the product of the `cfweight()` expression and the ordinary Stata weights if `cfweight()` is specified and the ordinary Stata weights are `fweight`s, and are equal to 1 otherwise.

The importance weights are used as described in **Methods and Formulas** below. They are equal to the ordinary Stata weights if these are specified and either `cluster()` or `cfweight()` is specified, are equal to the ordinary Stata weights if neither of these two options is specified and the ordinary Stata weights are specified as `pweight`s or `iweight`s, and are equal to 1 otherwise.

### 1.4 Saved results

`somersd` saves the following in `e()`:

| | | | |
|---|---|---|---|
| Scalars | | | |
| `e(N)` | number of observations | `e(N_clust)` | number of clusters |
| `e(df_r)` | residual degrees of freedom | `e(denominator)` | common denominator |
| `e(depvarsum)` | sum of $X$–variable in estimation sample | | |
| Macros | | | |
| `e(cmd)` | `somersd` | `e(cmdline)` | command as typed |
| `e(param)` | parameter (`somersd` or `taua`) | `e(parmlab)` | parameter label in output |
| `e(tdist)` | `tdist` if specified | `e(depvar)` | name of $X$–variable |
| `e(clustvar)` | name of cluster variable | `e(vcetype)` | title used to label standard error |
| `e(wtype)` | weight type | `e(wexp)` | weight expression |
| `e(cfweight)` | `cfweight()` expression | `e(funtype)` | `funtype()` option |
| `e(wstrata)` | `wstrata()` option | `e(bstrata)` | `bstrata()` option |
| `e(predict)` | program called by `predict` (`somers_p`) | `e(transf)` | transformation specified by `transf()` |
| `e(tranlab)` | transformation label in output | `e(properties)` | `b V` |
| Matrices | | | |
| `e(b)` | coefficient vector | `e(V)` | variance–covariance matrix of the estimators |
| Functions | | | |
| `e(sample)` | marks estimation sample | | |

Note that (confusingly) `e(depvar)` is the $X$–variable, or predictor variable, in the conventional terminology for defining Somers' $D$. `somersd` is also different from most estimation commands in that its results are not designed to be used by `predict`. If the user tries to do so, then the program `somers_p` is called, and tells the user that `predict` should not be used after `somersd`. The scalar `e(denominator)` contains the common denominator used in calculating the Somers' $D$ or Kendall's $\tau_a$ statistics.

## 2 Methods and formulas

The population value of Kendall's $\tau_a$ (Kendall and Gibbons, 1990) is traditionally defined as

$$\tau_{XY} = E\left[\mathrm{sign}(X_1 - X_2)\,\mathrm{sign}(Y_1 - Y_2)\right], \tag{1}$$

where $(X_1, Y_1)$ and $(X_2, Y_2)$ are bivariate random variables sampled independently from the same population, and $E[\cdot]$ denotes expectation. This definition can be generalized to possibly left– or right–censored and/or stratified and/or clustered and/or weighted data as follows. Suppose that 4–variate observations $(X_i, R_i, Y_i, S_i)$ are sampled from an arbitrary population, using an arbitrary sampling scheme. The $R_i$ are censorship indicators for the corresponding $X_i$, and the $S_i$ are censorship indicators for the corresponding $Y_i$. These censorship indicators are negative in the case of left–censorship (where the "true" value of the indicated variable is known to be equal to or less than its recorded value), positive in the case of right–censorship (in which the "true" value of the indicated variable is known to be equal to or greater than its recorded value), and zero in the case of non–censorship (in which the "true" value is known to be equal to the recorded value). We define a "censored sign difference" for two values $u$ and $v$, with respective censorship indicators $p$ and $q$, as

$$\mathrm{csign}(u, p, v, q) = \begin{cases} 1, & u > v \text{ and } p \geq 0 \geq q \\ -1, & u < v \text{ and } p \leq 0 \leq q \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Given two observations $(X_i, R_i, Y_i, S_i)$ and $(X_j, R_j, Y_j, S_j)$, we will denote the product of $\mathrm{csign}(X_i, R_i, X_j, R_j)$ and $\mathrm{csign}(Y_i, S_i, Y_j, S_j)$ as the concordance–discordance difference for the two observations, and say that the two observations are concordant if this product is 1, discordant if the product is $-1$, and neither concordant nor discordant if the product is zero. We can now redefine Kendall's $\tau_a$ as

$$\tau_{XY} = E\left[\mathrm{csign}(X_i, R_i, X_j, R_j)\,\mathrm{csign}(Y_i, S_i, Y_j, S_j)\right], \tag{3}$$

or (in words) as the mean concordance–discordance difference. This expectation can be defined using weights specific to the observations and/or restrictions to subsets of pairs of observations, defined in terms of the sampling scheme.

The population value of Somers' $D$ (Somers, 1962) is defined as

$$D_{YX} = \frac{\tau_{XY}}{\tau_{XX}}. \tag{4}$$

Therefore, $\tau_{XY}$ is the difference between two probabilities, namely the probability that the larger of the two $X$–values is associated with the larger of the two $Y$–values and the probability that the larger $X$–value is associated with the smaller $Y$–value. $D_{YX}$ is the difference between the two corresponding *conditional* probabilities, given that the two $X$–values are known to be unequal. Somers' $D$ is related to Harrell's $c$ index by the formula $D = 2c - 1$ (see Harrell *et al.*, 1982 and Harrell *et al.*, 1996). Kendall's $\tau_a$ is the covariance between $\mathrm{csign}(X_i, R_i, X_j, Y_j)$ and $\mathrm{csign}(Y_i, S_i, Y_j, S_j)$, whereas Somers' $D$ is the regression coefficient of $\mathrm{csign}(Y_i, S_i, Y_j, S_j)$ with respect to $\mathrm{csign}(X_i, R_i, X_j, R_j)$. The correlation coefficient between $\mathrm{csign}(X_i, R_i, X_j, R_j)$ and $\mathrm{csign}(Y_i, S_i, Y_j, S_j)$ is known as Kendall's $\tau_b$, and is equal to the geometric mean of the absolute values of $D_{YX}$ and $D_{XY}$ multiplied by their common sign.

Given a sample of observations $(X_i, R_i, Y_i, S_i)$, we may estimate and test the population values of Kendall's $\tau_a$ and Somers' $D$ by the corresponding sample statistics $\hat{\tau}_{XY}$ and $\hat{D}_{YX}$. These are commonly known as "non–parametric" statistics, even though $\tau_{XY}$ and $D_{YX}$ are parameters. The two Wilcoxon rank–sum tests (see [R] `ranksum` and [R] `signrank`) both test hypotheses predicting $D_{YX} = 0$. The two–sample rank–sum test represents the case where $X$ is a binary variable indicating membership of one of two sub–populations. If the binary $X$–variable indicates that a patient has a disease, and the $Y$–variable is a continuous diagnostic test indicator with high values indicating a high probability that the patient has the disease, then the area $A$ under the receiver operating characteristic (ROC) curve, or sensitivity–specificity curve, is linked to Somers' $D$ by the relation $D_{YX} = 2A - 1$. (See [R] **roc** or Hanley and McNeil, 1982.) The matched–pairs rank–sum test represents the case where there are paired data $(W_{i1}, W_{i2})$, such that $X_i = \mathrm{sign}(W_{i1} - W_{i2})$, and $Y_i = |W_{i1} - W_{i2}|$. Kendall's $\tau_a$ is usually tested on "continuous" data, using `ktau` (see [R] `spearman`).

### 2.1 Motivation for confidence intervals

There are several reasons for preferring confidence intervals to $P$–values alone:

1. Non–statisticians often quote a “non–significant” result for a “non–parametric” test and argue as if they have “proved” a null hypothesis, when a confidence interval would show a wide range of other hypotheses which *also* fit the data.
2. In the case of continuous bivariate data, there is a correspondence between Kendall's $\tau_a$ and the better–known Pearson's correlation coefficient $\rho$, known as Greiner's relation (Kendall and Gibbons, 1990). This states that
$$\rho = \sin\left(\frac{\pi}{2}\tau_a\right), \tag{5}$$
and holds if the joint distribution of $X$ and $Y$ is bivariate normal. Under this relation, Kendall's $\tau_a$–values of 0, $\pm\frac{1}{3}$, $\pm\frac{1}{2}$ and $\pm 1$ correspond to Pearson's correlations of 0, $\pm\frac{1}{2}$, $\pm\frac{1}{\sqrt{2}}$ and $\pm 1$, respectively. A similar correspondence is likely to hold in a wider range of continuous bivariate distributions (Kendall, 1949; Newson, 2002).
3. Kendall's $\tau_a$ has the desirable property that a larger $\tau_a$ cannot be secondary to a smaller $\tau_a$. That is to say, if a positive $\tau_{XY}$ is caused entirely by a monotonic positive relationship of both variables with a third variable $W$, then $\tau_{WX}$ and $\tau_{WY}$ must both be greater than $\tau_{XY}$. If we can show that $\tau_{XY} - \tau_{WY} > 0$ (or, equivalently, that $D_{XY} - D_{WY} > 0$), then this implies that the correlation between $X$ and $Y$ is not caused entirely by the influence of $W$.

To understand the third point, assume that data points $(W_i, X_i, Y_i, S_i)$ are sampled independently from a common population, with discrete probability mass function $f_{W,X,Y,S}(\cdot,\cdot,\cdot,\cdot)$ and marginal probability mass function $f_{W,X}(\cdot,\cdot)$. We assume that the $S_i$ are censorship indicators for the corresponding outcome variables $Y_i$, and that the $X_i$ and $W_i$ are alternative uncensored predictors. Define the conditional expectation

$$Z(w_1, x_1, w_2, x_2) = E\left[\,\mathrm{csign}(Y_2, S_2, Y_1, S_1)\,|\, W_1 = w_1, X_1 = x_1, W_2 = w_2, X_2 = x_2\,\right] \tag{6}$$

for any $w_1$ and $w_2$ in the range of $W$–values and any $x_1$ and $x_2$ in the range of $X$–values. If we state that the positive relationship between $X_i$ and $Y_i$ is caused entirely by a monotonic positive relationship between both variables and $W_i$, then that is equivalent to stating that

$$Z(w_1, x_1, w_2, x_2) \geq 0 \tag{7}$$

whenever $w_1 \leq w_2$ and $x_2 \leq x_1$. However, the difference between the two $\tau_a$ coefficients is

$$\begin{aligned}\tau_{WY} - \tau_{XY} =& 2\sum_{w}\sum_{x_2<x_1} f_{W,X}(w,x_1)\, f_{W,X}(w,x_2)\, Z(w,x_1,w,x_2)\\ &+ 2\sum_{x}\sum_{w_1<w_2} f_{W,X}(w_1,x)\, f_{W,X}(w_2,x)\, Z(w_1,x,w_2,x)\\ &+ 4\sum_{w_1<w_2}\sum_{x_2<x_1} f_{W,X}(w_1,x_1)\, f_{W,X}(w_2,x_2)\, Z(w_1,x_1,w_2,x_2).\end{aligned} \tag{8}$$

This difference must be non–negative whenever the inequality (7) applies. In particular, if the distribution of the $W_i$ and $X_i$ is nearly continuous, then the difference (8) will be dominated by the third term, representing discordant $(W_i, X_i)$–pairs.

### 2.2 Estimation formulas for Somers' $D$ and Kendall's $\tau_a$

Somers' $D$ and Kendall's $\tau_a$, in their various forms, can be expressed as ratios of sample means, Hoeffding $U$–statistics or von Mises $V$–statistics, depending on the functional type specified by the `funtype()` option. `somersd` works by jackknifing the original means, $U$–statistics and $V$–statistics (Arvesen, 1969), and by using Taylor polynomials to derive variances for the ratios. Normalizing and/or variance–stabilizing transformations may then be applied.

We assume the general case where the observations are clustered, which becomes the familiar unclustered case when there is one observation per cluster, and that there are $N$ clusters in the sample, sampled from a common population. We assume that there are one or more indexed $W$–strata (defaulting to one all–inclusive $W$–stratum if `wstrata()` is not specified). Two slightly different versions of the notation will be used, depending on whether or not the user has specified $B$–strata using the `bstrata()` option.

If there are no $B$–strata, then we define $w_{fgi}$, $X_{fgi}$, $Y_{fgi}$, $R_{fgi}$ and $S_{fgi}$ to be the importance weight, $X$–value, $Y$–value, $X$–censorship indicator and $Y$–censorship indicator, respectively, for the $i$th observation belonging to the $g$th $W$–stratum in the $f$th cluster. Not every possible index combination $fgi$ will correspond to an observation, so all summation over index combinations will be over index combinations corresponding to an observation. For index combinations $fgi$ and $jkm$ corresponding to observations, we can define

$$\begin{aligned} v_{fgi,jkm} &= w_{fgi} w_{jkm}, \\ t^{(XY)}_{fgi,jkm} &= v_{fgi,jkm} \operatorname{csign}(X_{fgi}, R_{fgi}, X_{jkm}, R_{jkm}) \operatorname{csign}(Y_{fgi}, S_{fgi}, Y_{jkm}, S_{jkm}). \end{aligned} \tag{9}$$

We will use the usual dot–substitution notation to define (for instance)

$$v_{fgi,jk.} = \sum_m v_{fgi,jkm}, \quad t^{(XY)}_{fgi,jk.} = \sum_m t^{(XY)}_{fgi,jkm}, \quad v_{fgi,j..} = \sum_k v_{fgi,jk.}, \quad t^{(XY)}_{fgi,j..} = \sum_k t^{(XY)}_{fgi,jk.}, \tag{10}$$

and any other sums over any other indices. For clusters $f$ and $j$, we define

$$\phi^{(V)}_{fj} = \sum_g v_{fg.,jg.}, \quad \phi^{(XY)}_{fj} = \sum_g t^{(XY)}_{fg.,jg.}. \tag{11}$$

In other words, $\phi^{(V)}_{fj}$ is the sum of pairwise importance weights, and $\phi^{(XY)}_{fj}$ is the sum of pairwise importance–weighted concordance–discordance differences, belonging to pairs of observations, in the same $W$–stratum, of which the first observation is in cluster $f$ and the second observation is in cluster $j$. The quantities $\phi^{(V)}_{fj}$ and $\phi^{(XY)}_{fj}$ are known as kernels in the terminology of Chapter 5 of Serfling (1980), and are defined for any pair of clusters.

If the user has defined $B$–strata, then we define the kernels $\phi^{(V)}_{fj}$ and $\phi^{(XY)}_{fj}$ by a slightly different formula. We define $w_{fghi}$, $X_{fghi}$, $Y_{fghi}$, $R_{fghi}$ and $S_{fghi}$ to be the importance weight, $X$–value, $Y$–value, $X$–censorship indicator and $Y$–censorship indicator, respectively, for the $i$th observation belonging to cluster $f$, $W$–stratum $g$ and $B$–stratum $h$. For index combinations $fghi$ and $jklm$ corresponding to observations, we define

$$\begin{aligned} v_{fghi,jklm} &= w_{fghi} w_{jklm}, \\ t^{(XY)}_{fghi,jklm} &= v_{fghi,jklm} \operatorname{csign}(X_{fghi}, R_{fghi}, X_{jklm}, R_{jklm}) \operatorname{csign}(Y_{fghi}, S_{fghi}, Y_{jklm}, S_{jklm}), \end{aligned} \tag{12}$$

and for clusters $f$ and $j$ we define

$$\phi^{(V)}_{fj} = \sum_g v_{fg..,jg..} - \sum_g \sum_h v_{fgh.,jgh.}, \quad \phi^{(XY)}_{fj} = \sum_g t^{(XY)}_{fg..,jg..} - \sum_g \sum_h t^{(XY)}_{fgh.,jgh.}. \tag{13}$$

This time, $\phi^{(V)}_{fj}$ is the sum of products of importance weights, and $\phi^{(XY)}_{fj}$ is the sum of importance–weighted concordance–discordance differences, belonging to pairs of observations, in the same $W$–stratum and different $B$–strata, of which the first observation is in cluster $f$ and the second observation is in cluster $j$. Note that, if the user has specified `bstrata(_n)`, then every observation is in its own $B$–stratum, and the second terms in the $\phi^{(V)}_{fj}$ and $\phi^{(XY)}_{fj}$ of (13) will then contain only pairs in which an observation is paired with itself.

The kernels $\phi^{(V)}_{fj}$ and $\phi^{(XY)}_{fj}$ of (11) or (13) can be “averaged” over their indices to produce parameters denoted as $V$ and $T_{XY}$, respectively. Kendall’s $\tau_a$ and Somers’ $D$ are defined as ratios of these “averages” by

$$\tau_{XY} = T_{XY}/V, \quad D_{YX} = T_{XY}/T_{XX} = \tau_{XY}/\tau_{XX}. \tag{14}$$

The way in which the kernels are averaged depends on the `funtype()` option. If the user specifies `funtype(wcluster)`, then $V$ and $T_{XY}$ are “within–cluster averages”. If the user specifies `funtype(bcluster)` (the default), then $V$ and $T_{XY}$ are “between–cluster averages”. If the user specifies `funtype(vonmises)`, then $V$ and $T_{XY}$ are “overall averages”. In all cases, we estimate the population parameters $V$ and $T_{XY}$ using sample statistics $\hat{V}$ and $\hat{T}_{XY}$ as point estimates, and estimate the sampling variances of these point estimates using a jackknife method, with pseudovalues denoted $\psi^{(V)}_j$ and $\psi^{(XY)}_j$ for the $j$th cluster.

If the user specifies `funtype(wcluster)`, then `somersd` estimates the parameters

$$V = E\left[\phi^{(V)}_{jj}\right], \quad T_{XY} = E\left[\phi^{(XY)}_{jj}\right]. \tag{15}$$

These functionals are population means of within–cluster kernels, and their point estimates are the corresponding sample means

$$\hat{V} = N^{-1}\sum_{j=1}^{N}\phi_{jj}^{(V)}, \quad \hat{T}_{XY} = N^{-1}\sum_{j=1}^{N}\phi_{jj}^{(XY)}, \tag{16}$$

and the jackknife pseudovalues for the $j$th cluster are given by

$$\psi_j^{(V)} = \phi_{jj}^{(V)}, \quad \psi_j^{(XY)} = \phi_{jj}^{(XY)}. \tag{17}$$

If the user has specified `funtype(bcluster)` (the default) or `funtype(vonmises)`, then `somersd` estimates the parameters

$$V = E\left[\phi_{fj}^{(V)}\right], \quad T_{XY} = E\left[\phi_{fj}^{(XY)}\right], \tag{18}$$

for $f \neq j$. These parameters are known as Hoeffding functionals if clusters $f$ and $j$ are assumed to be sampled without replacement, and as von Mises functionals if clusters $f$ and $j$ are assumed to be sampled with replacement. (If the population from which the clusters are sampled is infinite, then the population Hoeffding functional is equal to the corresponding population von Mises functional.)

If the user specifies `funtype(bcluster)`, or does not specify a `funtype()` option, then the point estimates of the population Hoeffding functionals are the corresponding sample Hoeffding functionals, or $U$–statistics in the terminology of Hoeffding (1948), defined as $\hat{V} = \hat{T}_{XY} = 0$ if $N = 1$, and otherwise as

$$\hat{V} = \frac{\phi_{..}^{(V)} - \sum_{j=1}^{N}\phi_{jj}^{(V)}}{N(N-1)}, \quad \hat{T}_{XY} = \frac{\phi_{..}^{(XY)} - \sum_{j=1}^{N}\phi_{jj}^{(XY)}}{N(N-1)}. \tag{19}$$

The jackknife pseudovalues for the $j$th cluster are given by $\psi_j^{(V)} = \psi_j^{(XY)} = 0$ if $N = 1$, by

$$\psi_j^{(V)} = \phi_{j.}^{(V)} - \phi_{jj}^{(V)}, \quad \psi_j^{(XY)} = \phi_{j.}^{(XY)} - \phi_{jj}^{(XY)} \tag{20}$$

if $N = 2$, and otherwise as

$$\begin{aligned}
\psi_j^{(V)} &= (N-1)^{-1}\left(\phi_{..}^{(V)} - \sum_{k=1}^{N}\phi_{kk}^{(V)}\right) - (N-2)^{-1}\left[\phi_{..}^{(V)} - \sum_{k=1}^{N}\phi_{kk}^{(V)} - 2\left(\phi_{j.}^{(V)} - \phi_{jj}^{(V)}\right)\right], \\
\psi_j^{(XY)} &= (N-1)^{-1}\left(\phi_{..}^{(XY)} - \sum_{k=1}^{N}\phi_{kk}^{(XY)}\right) - (N-2)^{-1}\left[\phi_{..}^{(XY)} - \sum_{k=1}^{N}\phi_{kk}^{(XY)} - 2\left(\phi_{j.}^{(XY)} - \phi_{jj}^{(XY)}\right)\right].
\end{aligned} \tag{21}$$

If the user specifies `funtype(vonmises)`, then the point estimates of the population von Mises functionals are the corresponding sample von Mises functionals, or $V$–statistics in the terminology of Chapter 5 of Serfling (1980). These are defined as

$$\hat{V} = N^{-2}\phi_{..}^{(V)}, \quad \hat{T}_{XY} = N^{-2}\phi_{..}^{(XY)}, \tag{22}$$

and the jackknife pseudovalues for the $j$th cluster are given by

$$\psi_j^{(V)} = \phi_{jj}^{(V)}, \quad \psi_j^{(XY)} = \phi_{jj}^{(XY)} \tag{23}$$

if $N = 1$, and otherwise by

$$\begin{aligned}
\psi_j^{(V)} &= N^{-1}\phi_{..}^{(V)} - (N-1)^{-1}\left(\phi_{..}^{(V)} - 2\phi_{j.}^{(V)} + \phi_{jj}^{(V)}\right), \\
\psi_j^{(XY)} &= N^{-1}\phi_{..}^{(XY)} - (N-1)^{-1}\left(\phi_{..}^{(XY)} - 2\phi_{j.}^{(XY)} + \phi_{jj}^{(XY)}\right).
\end{aligned} \tag{24}$$

Note that the estimates and jackknife pseudovalues of formulas (15) to (24) can all be expressed in terms of the $\phi_{jj}^{(V)}$, $\phi_{j.}^{(V)}$, $\phi_{jj}^{(XY)}$ and $\phi_{j.}^{(XY)}$. Newson (2006a) introduced an algorithm to calculate these quantities, using binary search trees, which requires an amount of computation time of order $N_{\mathrm{obs}} \log N_{\mathrm{obs}}$, where $N_{\mathrm{obs}}$ is the total number of observations. A version of this algorithm is used by `somersd`, unless the user specifies the `notree` option, in which case `somersd` uses a trivial algorithm, which compares all pairs of observations and requires an amount of time quadratic in $N_{\mathrm{obs}}$. The difference in performance can be spectacular in large datasets ($N_{\mathrm{obs}} > 1000$).

The parameters we really want to estimate are Kendall's $\tau_a$ and/or Somers' $D$, defined by (14). These formulas are equal to the familiar formulas (3) and (4) if each cluster contains one observation with an importance weight of one. To estimate them, we use the jackknife method on $V$ and $T_{XY}$, and use appropriate Taylor polynomials. `somersd` calculates correlation measures for a single variable $X$ with a set of $Y$–variates $(Y^{(1)}, \ldots, Y^{(p)})$. (The $X$–variate may have a censorship indicator $R$, and the $Y$–variates may have censorship indicators $(S^{(1)}, \ldots, S^{(p)})$.) It calculates, in the first instance, the covariance matrix for $\hat{V}$, $\hat{T}_{XX}$, and $\hat{T}_{XY^{(i)}}$ for $1 \le i \le p$. This is done using the jackknife influence matrix $\Upsilon$, which has $N$ rows labelled by the cluster subscripts, and $p+2$ columns labelled (in Stata fashion) by the names $V$, $X$, and $Y^{(i)}$ for $1 \le i \le p$. It is defined by

$$\Upsilon\left[j, V\right] = \psi_j^{(V)} - \bar{\psi}^{(V)}, \quad \Upsilon\left[j, X\right] = \psi_j^{(XX)} - \bar{\psi}^{(XX)}, \quad \Upsilon\left[j, Y^{(i)}\right] = \psi_j^{(XY^{(i)})} - \bar{\psi}^{(XY^{(i)})}, \tag{25}$$

where the quantities

$$\bar{\psi}^{(V)} = N^{-1} \sum_{k=1}^{N} \psi_k^{(V)}, \quad \bar{\psi}^{(XX)} = N^{-1} \sum_{k=1}^{N} \psi_k^{(XX)}, \quad \bar{\psi}^{(XY^{(i)})} = N^{-1} \sum_{k=1}^{N} \psi_k^{(XY^{(i)})} \tag{26}$$

are the mean pseudovalues. (These mean pseudovalues are equal to the corresponding point estimates unless `funtype(vonmises)` is specified, in which case the mean pseudovalue is equal to the corresponding Hoeffding $U$–statistic.) The jackknife covariance matrix is equal to

$$\hat{C} = \left[N(N-1)\right]^{-1} \Upsilon' \Upsilon. \tag{27}$$

The estimates for Kendall's $\tau_a$ and Somers' $D$, for variables $Y$ and $X$, are defined by

$$\hat{\tau}_{XY} = \hat{T}_{XY} / \hat{V}, \quad \hat{D}_{YX} = \hat{T}_{XY} / \hat{T}_{XX}, \tag{28}$$

unless the denominators of these expressions are zero, in which case the numerators must also be zero, and `somersd` therefore sets the estimates and their covariances to zero. (The returned result `e(denominator)` contains $\hat{V}$ if `taua` is specified, and $\hat{T}_{XX}$ otherwise.) If the denominator is nonzero, then the covariance matrix is defined using Taylor polynomials. In the case of Somers' $D$, we define the $p \times (p+2)$ matrix of estimated derivatives $\hat{\Gamma}^{(D)}$, whose rows are labelled by the names $Y^{(1)}, \ldots, Y^{(p)}$, and whose columns are labelled by $V, X, Y^{(1)}, \ldots, Y^{(p)}$. This matrix is defined by

$$\begin{aligned} \hat{\Gamma}^{(D)}\left[Y^{(i)}, X\right] &= \frac{\partial \hat{D}_{Y^{(i)}X}}{\partial \hat{T}_{XX}} &= -\frac{\hat{T}_{XY^{(i)}}}{\hat{T}_{XX}^2}, \\ \hat{\Gamma}^{(D)}\left[Y^{(i)}, Y^{(i)}\right] &= \frac{\partial \hat{D}_{Y^{(i)}X}}{\partial \hat{T}_{XY^{(i)}}} &= \frac{1}{\hat{T}_{XX}}, \end{aligned} \tag{29}$$

all other entries being 0. In the case of Kendall's $\tau_a$, we define a $(p+1) \times (p+2)$ matrix of estimated derivatives $\hat{\Gamma}^{(\tau)}$, whose rows are labelled by $X, Y^{(1)}, \ldots, Y^{(p)}$, and whose columns are labelled by $V, X, Y^{(1)}, \ldots, Y^{(p)}$. This matrix is defined by

$$\begin{aligned} \hat{\Gamma}^{(\tau)}\left[X, V\right] &= \frac{\partial \hat{\tau}_{XX}}{\partial \hat{V}} &= -\frac{\hat{T}_{XX}}{\hat{V}^2}, \\ \hat{\Gamma}^{(\tau)}\left[X, X\right] &= \frac{\partial \hat{\tau}_{XX}}{\partial \hat{T}_{XX}} &= \frac{1}{\hat{V}}, \\ \hat{\Gamma}^{(\tau)}\left[Y^{(i)}, V\right] &= \frac{\partial \hat{\tau}_{XY^{(i)}}}{\partial \hat{V}} &= -\frac{\hat{T}_{XY^{(i)}}}{\hat{V}^2}, \\ \hat{\Gamma}^{(\tau)}\left[Y^{(i)}, Y^{(i)}\right] &= \frac{\partial \hat{\tau}_{XY^{(i)}}}{\partial \hat{T}_{XY^{(i)}}} &= \frac{1}{\hat{V}}, \end{aligned} \tag{30}$$

all other entries again being 0. The estimated dispersion matrices of the Somers' $D$ and $\tau_a$ estimates are therefore $\hat{C}^{(D)}$ and $\hat{C}^{(\tau)}$, respectively, defined by

$$\hat{C}^{(D)} = \hat{\Gamma}^{(D)}\,\hat{C}\,\hat{\Gamma}^{(D)\,\prime}, \quad \hat{C}^{(\tau)} = \hat{\Gamma}^{(\tau)}\,\hat{C}\,\hat{\Gamma}^{(\tau)\,\prime}. \tag{31}$$

## 2.3 Transformations

The `transf()` option offers a choice of transformations. Since these are available both for Somers' $D$ and for Kendall's $\tau_a$, we will denote the original estimate as $\theta$ (which can stand for $D$ or $\tau$) and the transformed estimate as $\zeta$. They are summarized below, together with their derivatives $d\zeta/d\theta$ and their inverses $\theta(\zeta)$.

| `transf()` | Transform name | $\zeta(\theta)$ | $d\zeta/d\theta$ | $\theta(\zeta)$ |
|---|---|---|---|---|
| `iden` | Untransformed | $\theta$ | $1$ | $\zeta$ |
| `z` | Fisher's $z$ | $\mathrm{arctanh}(\theta) =$<br>$\frac{1}{2}\log[(1+\theta)/(1-\theta)]$ | $\left(1-\theta^2\right)^{-1}$ | $\tanh(\zeta) =$<br>$[\exp(2\zeta)-1]/[\exp(2\zeta)+1]$ |
| `asin` | Daniels' arcsine | $\arcsin(\theta)$ | $\left(1-\theta^2\right)^{-1/2}$ | $\sin(\zeta)$ |
| `rho` | Greiner's $\rho$ | $\sin(\frac{\pi}{2}\theta)$ | $\frac{\pi}{2}\cos(\frac{\pi}{2}\theta)$ | $(2/\pi)\arcsin(\zeta)$ |
| `zrho` | Greiner's $\rho$<br>($z$-transformed) | $\mathrm{arctanh}[\sin(\frac{\pi}{2}\theta)]$ | $\frac{\pi}{2}\cos(\frac{\pi}{2}\theta)[1-\sin(\frac{\pi}{2}\theta)^2]^{-1}$ | $(2/\pi)\arcsin[\tanh(\zeta)]$ |
| `c` | Harrell's $c$ | $(\theta+1)/2$ | $1/2$ | $2\zeta-1$ |

Note that all of these expressions are defined for $\theta = 0$, but some are undefined for $\theta = 1$ or $\theta = -1$, and, in those cases, `somersd` enters a substitute $\theta$–argument very close to 1 or $-1$. (The exact limits beyond which substitute $\theta$–arguments are used are specified in a Mata function `_somdtransf()`, which calculates the transformations and their derivatives, and they are listed in the on–line documentation for this Mata function.) If `transf()` is specified, then `somersd` displays and saves the transformed estimates and their estimated covariance, instead of the untransformed versions. If $\hat{C}^{(\theta)}$ is the covariance matrix for the untransformed estimates given by (31), and $\hat{\Gamma}^{(\zeta)}$ is the diagonal matrix whose diagonal entries are the $d\zeta/d\theta$ estimates specified in the table, then the transformed parameter and its covariance matrix are

$$\hat{\zeta} = \zeta(\hat{\theta}), \quad \hat{C}^{(\zeta)} = \hat{\Gamma}^{(\zeta)}\,\hat{C}^{(\theta)}\,\hat{\Gamma}^{(\zeta)\,\prime}. \tag{32}$$

Fisher's $z$–transform was originally recommended for the Pearson correlation coefficient by Fisher (1921) (see also Gayen (1951)), but Edwardes (1995) recommended it specifically for Somers' $D$ on the basis of simulation studies. Daniels' arcsine was suggested as a normalizing transform in Daniels and Kendall (1947). If `transf(z)` or `transf(asin)` is specified, then `somersd` prints asymmetric confidence intervals for the untransformed $D$ or $\tau_a$ parameters, calculated from symmetric confidence intervals for the transformed parameters using the inverse function $\theta(\zeta)$. (This feature corresponds to the `eform` option of other estimation commands.) Greiner's $\rho$ (Kendall and Gibbons, 1990) is based on the relation (5), and is designed to estimate the Pearson correlation coefficient corresponding to the measured $\tau_a$. If `transf(zrho)` is specified, then `somersd` prints asymmetric confidence intervals for the untransformed Greiner's $\rho$, using the inverse $z$–transform on symmetric confidence intervals for the $z$–transformed Greiner's $\rho$. Harrell's $c$ is usually a reparameterization of Somers' $D$, and is recommended in Harrell *et al.* (1982) and Harrell *et al.* (1996) as a general measure of the predictive power of a prognostic score arising from a medical test.

# 3 Examples

We present 4 examples, using datasets provided by Stata Press at *http://www.stata-press.com/data/*.

## 3.1 Somers' $D$ in the `auto` data

In the `auto` data, we compare US cars with foreign cars regarding weight and fuel efficiency. First, we use `ranksum` to give significance tests without confidence intervals:

```
. ranksum mpg,by(foreign)
Two-sample Wilcoxon rank-sum (Mann-Whitney) test
     foreign |      obs    rank sum    expected
-------------+---------------------------------
    Domestic |       52      1688.5        1950
     Foreign |       22      1086.5         825
-------------+---------------------------------
    combined |       74        2775        2775
```

```
unadjusted variance     7150.00
adjustment for ties      -36.95
                     ----------
adjusted variance       7113.05
Ho: mpg(foreign==Domestic) = mpg(foreign==Foreign)
             z =  -3.101
    Prob > |z| =   0.0019
    Exact Prob =   0.0016
. ranksum weight,by(foreign)
Two-sample Wilcoxon rank-sum (Mann-Whitney) test
     foreign |      obs    rank sum    expected
-------------+---------------------------------
    Domestic |       52      2379.5        1950
     Foreign |       22       395.5         825
-------------+---------------------------------
    combined |       74        2775        2775
unadjusted variance     7150.00
adjustment for ties       -1.06
                     ----------
adjusted variance       7148.94
Ho: weight(foreign==Domestic) = weight(foreign==Foreign)
             z =   5.080
    Prob > |z| =   0.0000
    Exact Prob =   0.0000
```

We note that US cars are typically heavier and travel fewer miles per gallon than foreign cars. For confidence intervals, we use somersd:

```
. somersd foreign mpg weight
Somers' D with variable: foreign
Transformation: Untransformed
Valid observations: 74
Symmetric 95% CI
------------------------------------------------------------------------------
             |              Jackknife
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   .4571678    .135146     3.38   0.001     .1922866    .7220491
      weight |  -.7508741   .0832485    -9.02   0.000    -.9140383    -.58771
------------------------------------------------------------------------------
```

We see that, given a randomly–chosen foreign car and a randomly–chosen US car, the foreign car is 46% more likely to travel more miles per gallon than the US car than *vice versa*, with confidence limits from 19% to 72% more likely. However, being foreign seems to be more reliable as a negative predictor of weight than as a positive predictor of “fuel efficiency”. We can use lincom to define confidence limits for the difference:

```
. lincom -weight-mpg
 ( 1)  - mpg - weight = 0
------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   .2937063   .0884397     3.32   0.001     .1203677    .4670449
------------------------------------------------------------------------------
```

The difference between Somers' $D$–values is positive. This indicates that, if there are two cars, one heavier and consuming fewer gallons per mile, the other lighter and consuming more gallons per mile, then the second is more likely to be foreign. So maybe 1970s US cars were not as wasteful as some people think, and were, if anything, more fuel–efficient for their weight than non–US cars at the time. Figure 1 illustrates this graphically. Data points are domestic cars (“D”) and foreign cars (“F”). A regression analysis could show the same thing, but Somers' $D$ shows it in stronger terms, without contentious assumptions such as linearity. (On the other hand, a regression model is more informative if its assumptions are true, so the two methods are mutually complementary.)

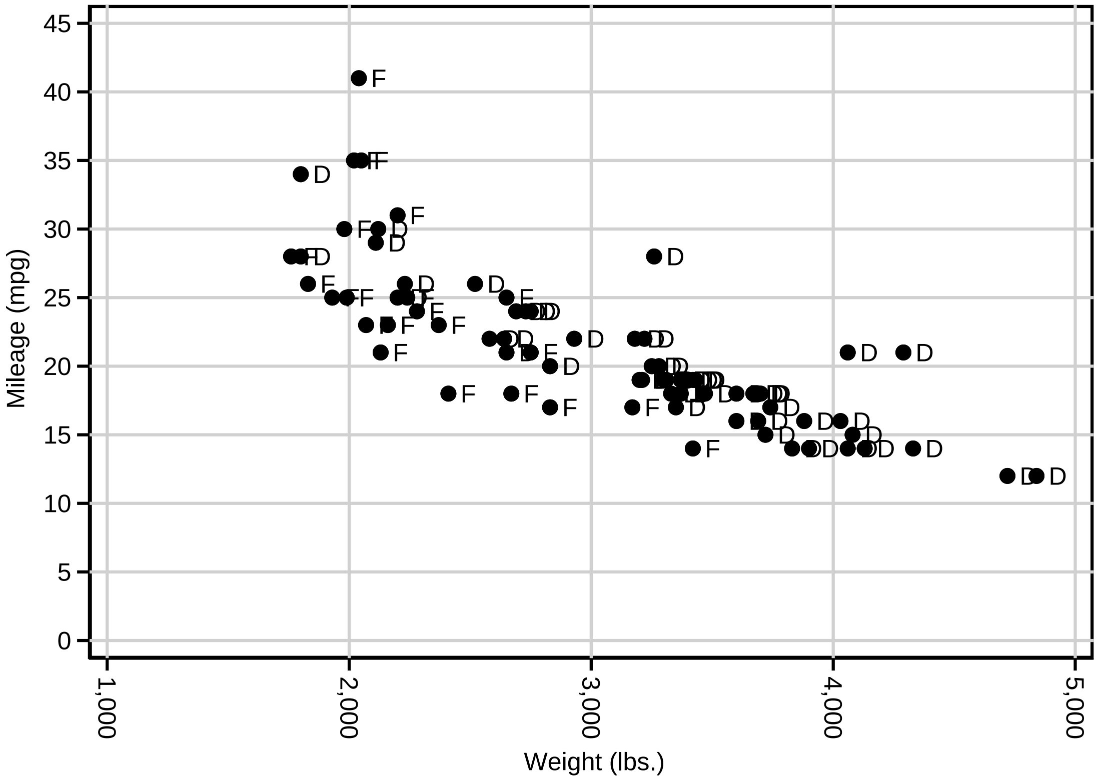

Figure 1. Mileage and weight in US cars (D) and non–US cars (F)

The confidence intervals for such high values of Somers' $D$ would probably be more reliable if we used the $z$–transform, recommended by Edwardes (1995). The results of this are as follows:

```
Somers' D with variable: foreign
Transformation: Fisher's z
Valid observations: 74
Symmetric 95% CI for transformed Somers' D
------------------------------------------------------------------------------
             |              Jackknife
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   .4937249   .1708551     2.89   0.004     .1588551    .8285947
      weight |  -.9749561   .1908547    -5.11   0.000    -1.349024   -.6008878
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Somers' D
               Somers_D     Minimum     Maximum
         mpg   .45716783   .15753219   .67972072
      weight  -.75087413  -.87382282  -.53768098
. lincom -weight-mpg
 ( 1)  - mpg - weight = 0
------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   .4812312   .1235452     3.90   0.000     .2390871    .7233753
------------------------------------------------------------------------------
```

Note that `somersd` gives not only symmetric confidence limits for the $z$–transformed Somers' $D$ estimates, but also the more informative asymmetric confidence limits for the untransformed Somers' $D$ estimates (corresponding to the `eform` option). The asymmetric confidence limits for the untransformed estimates are closer to zero than the symmetric confidence limits for the untransformed estimates in the previous output, and are probably more realistic. The output to `lincom` gives confidence limits for the difference between $z$–transformed Somers' $D$ values. This difference is expressed in $z$–units, but must, of course, be in the same direction as the difference between untransformed Somers' $D$ values. The conclusions are similar.

### 3.2 Kendall's $\tau_a$ in the auto data

In this example, we demonstrate Kendall's $\tau_a$ by comparing weight (pounds) and displacement (cubic inches) as predictors of fuel efficiency (miles per gallon). We first use ktau to carry out significance tests with no confidence limits:

```
. ktau mpg mpg
  Number of obs =      74
Kendall's tau-a =       0.9471
Kendall's tau-b =       1.0000
Kendall's score =    2558
    SE of score =     212.989   (corrected for ties)
Test of Ho: mpg and mpg are independent
     Prob > |z| =       0.0000  (continuity corrected)
. ktau mpg weight
  Number of obs =      74
Kendall's tau-a =      -0.6857
Kendall's tau-b =      -0.7059
Kendall's score =   -1852
    SE of score =     213.605   (corrected for ties)
Test of Ho: mpg and weight are independent
     Prob > |z| =       0.0000  (continuity corrected)
. ktau mpg displacement
  Number of obs =      74
Kendall's tau-a =      -0.5942
Kendall's tau-b =      -0.6257
Kendall's score =   -1605
    SE of score =     212.850   (corrected for ties)
Test of Ho: mpg and displacement are independent
     Prob > |z| =       0.0000  (continuity corrected)
```

We then use somersd (with the taua option and the $z$–transform) to compute the same statistics with confidence limits. Note that somersd also outputs the $\tau_a$ of mpg with mpg, which is simply the probability that two independently sampled mpg–values are not equal.

```
. somersd mpg weight displacement,taua
Kendall's tau-a with variable: mpg
Transformation: Untransformed
Valid observations: 74
Symmetric 95% CI
------------------------------------------------------------------------------
             |              Jackknife
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   .9470566   .0077145   122.76   0.000       .9319366  .9621767
      weight |   -.685672   .0445194   -15.40   0.000      -.7729283 -.5984156
displacement |  -.5942244   .0601971    -9.87   0.000      -.7122085 -.4762403
------------------------------------------------------------------------------
```

We can use lincom to compare the two predictors and test whether smaller and heavier cars travel fewer miles per gallon than larger and lighter cars. This seems to be the case, as weight is a more negative predictor of mpg than displacement:

```
. lincom weight-displacement
 ( 1)  weight - displacement = 0
------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.0914476   .0459194    -1.99   0.046      -.1814479 -.0014473
------------------------------------------------------------------------------
```

We demonstrate the cluster option using the variable manuf, equal to the first word of make, to denote manufacturer. This analysis assumes that we are sampling from the population of car manufacturers, rather than from the population of car models. The results are as follows:

```
. somersd mpg weight displacement,taua tr(z)
Kendall's tau-a with variable: mpg
Transformation: Fisher's z
Valid observations: 74
Symmetric 95% CI for transformed Kendall's tau-a
```

```
------------------------------------------------------------------------------
             |              Jackknife
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   1.802426   .0748368    24.08   0.000     1.655748    1.949103
      weight |  -.8397412    .084022    -9.99   0.000    -1.004421   -.6750612
displacement |  -.6841711    .093055    -7.35   0.000    -.8665556   -.5017866
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Kendall's tau-a
                   Tau_a     Minimum     Maximum
         mpg   .94705665   .92964223   .96024957
      weight  -.68567197  -.76344472  -.58829928
displacement  -.59422436  -.69961991  -.46352103
. lincom weight-displacement
 ( 1)  weight - displacement = 0
------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.1555701   .0742717    -2.09   0.036    -.3011399   -.0100003
------------------------------------------------------------------------------
```

Note that, in contrast to the case of most estimation commands, the `cluster` option affects the estimates as well as their confidence limits. This is because, in default, the clustered estimates are calculated only from between–cluster comparisons, in this case pairs of car models from different manufacturers. However, we can alter this by specifying `funtype(vonmises) bstrata(_n)`, in which case the estimates will be as before, but the confidence limits will be different:

```
. somersd mpg weight displacement,taua tr(z) cluster(manuf) ///
>   funtype(vonmises) bstrata(_n)
Von Mises Kendall's tau-a with variable: mpg
Transformation: Fisher's z
Between strata defined by: _n
Valid observations: 74
Number of clusters: 23
Symmetric 95% CI for transformed Kendall's tau-a
                                 (Std. Err. adjusted for 23 clusters in manuf)
------------------------------------------------------------------------------
             |              Jackknife
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   1.802426   .0813496    22.16   0.000     1.642983    1.961868
      weight |  -.8397412   .0905411    -9.27   0.000    -1.017199   -.6622839
displacement |  -.6841711   .0940933    -7.27   0.000    -.8685905   -.4997517
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Kendall's tau-a
                   Tau_a     Minimum     Maximum
         mpg   .94705665   .92788848   .96123213
      weight  -.68567197  -.76872301  -.57988136
displacement  -.59422436  -.70065729  -.46192188
. lincom weight-displacement
 ( 1)  weight - displacement = 0
------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.1555701   .0817275    -1.90   0.057     -.315753    .0046128
------------------------------------------------------------------------------
```

Alternatively, we might want to compare the power of weight and displacement to predict mileage when comparing cars made by the same manufacturer. To do this, we use the options `funtype(wcluster) bstrata(_n)`:

```
. somersd mpg weight displacement,taua tr(z) cluster(manuf) ///
>   funtype(wcluster) bstrata(_n)
Within-cluster Kendall's tau-a with variable: mpg
Transformation: Fisher's z
Between strata defined by: _n
Valid observations: 74
Number of clusters: 23
Symmetric 95% CI for transformed Kendall's tau-a
                                 (Std. Err. adjusted for 23 clusters in manuf)
------------------------------------------------------------------------------
```

```
             |               Jackknife
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   1.396604   .1289473    10.83   0.000     1.143872    1.649336
      weight |   -.852374   .1701861    -5.01   0.000    -1.185933   -.5188155
displacement |  -.4992644    .070216    -7.11   0.000    -.6368852   -.3616436
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Kendall's tau-a
                   Tau_a     Minimum     Maximum
         mpg   .88461538   .81571376   .92876647
      weight  -.69230769  -.82931315  -.47678526
displacement  -.46153846  -.56277501  -.34666094
. lincom weight-displacement
 ( 1)  weight - displacement = 0
------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.3531096   .1551786    -2.28   0.023    -.6572541   -.0489652
------------------------------------------------------------------------------
```

We see that, once again, both weight and displacement are negative predictors of mileage, and that weight is a more negative predictor than displacement.

Suppose that we are writing for an audience more familiar with Pearson's correlation than with Kendall's $\tau_a$. To estimate the Pearson correlations corresponding to our $\tau_a$ coefficients, we use the `zrho` transform. The results are as follows:

```
. somersd mpg weight displacement,taua tr(zrho)
Kendall's tau-a with variable: mpg
Transformation: z-transform of Greiner's rho
Valid observations: 74
Symmetric 95% CI for transformed Greiner's rho
------------------------------------------------------------------------------
             |               Jackknife
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   3.179521   .1458796    21.80   0.000     2.893602    3.465439
      weight |  -1.378273   .1475561    -9.34   0.000    -1.667478   -1.089069
displacement |  -1.108838    .158893    -6.98   0.000    -1.420262   -.7974132
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Greiner's rho
                     Rho     Minimum     Maximum
         mpg   .99654393   .99388566   .99804762
      weight  -.88056403  -.93121746  -.79653796
displacement  -.80365118  -.88965364  -.66258811
```

The $\tau_a$ of -0.59 between displacement and fuel efficiency (from the unclustered output) is seen to correspond to a more impressive Pearson correlation of -0.80. The estimated Greiner's $\rho$ is probably less likely to be oversensitive to outliers than the usual Pearson coefficient.

### 3.3 Harrell's $c$ and Somers' $D$ in the `drugtr` data

In this example, we demonstrate the `cenind()` option with a simple set of survival data distributed by Stata Press, with 1 observation per subject in a drug trial and data on treatment, age and survival time. We first load the data, then tabulate the treatment variable `drug`, then define the new variables `youth` (representing number of years to the subject's 100th birthday) and `censind` (a censorship indicator equal to 0 for subjects who died and to 1 for subjects whose survival time is right–censored):

```
. use http://www.stata-press.com/data/r9/drugtr, clear
(Patient Survival in Drug Trial)
. tab drug, m
  Drug type |
(0=placebo) |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |         20       41.67       41.67
          1 |         28       58.33      100.00
------------+-----------------------------------
      Total |         48      100.00
. gene youth=100-age
. gene byte censind=1-died
. tab died censind, m
      1 if |
   patient |       censind
      died |         0          1 |     Total
-----------+----------------------+----------
         0 |         0         17 |        17
         1 |        31          0 |        31
-----------+----------------------+----------
     Total |        31         17 |        48
```

We then use somersd to estimate the Harrell's $c$ parameters of active treatment and youth with respect to survival time:

```
. somersd studytime drug youth, tr(c) cenind(censind)
Somers' D with variable: studytime
Transformation: Harrell's c
Valid observations: 48
Symmetric 95% CI for Harrell's c
------------------------------------------------------------------------------
             |              Jackknife
   studytime |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .7275986   .0367931    19.78   0.000     .6554855    .7997117
       youth |   .6415771   .0528314    12.14   0.000     .5380295    .7451246
------------------------------------------------------------------------------
. lincom drug-youth
 ( 1)  drug - youth = 0
------------------------------------------------------------------------------
   studytime |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   .0860215   .0618354     1.39   0.164    -.0351736    .2072166
------------------------------------------------------------------------------
```

We see that active drug treatment and youth are both positive survival indicators, as they both have values of Harrell's $c$ greater than 0.5. (See Harrell *et al.*, 1982 and Harrell *et al.*, 1996 for more information on Harrell's $c$.) However, when we use lincom to estimate the difference between the two Harrell's $c$ parameters (equal to half the difference between the corresponding Somers' $D$ parameters), we find that the confidence interval for the difference includes zero. Based on this difference alone, we cannot state that the active treatment is a more or less positive predictor than being young. *However*, we can split the sample into 3 age tertiles, and estimate pooled, stratified Harrell's $c$ values for youth and treatment (based only on comparisons within age tertiles), and their difference:

```
. xtile agegp=age, n(3)
. tab agegp, m
3 quantiles |
     of age |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |         18       37.50       37.50
          2 |         16       33.33       70.83
          3 |         14       29.17      100.00
------------+-----------------------------------
      Total |         48      100.00
. somersd studytime drug youth, tr(c) cenind(censind) wstrata(agegp)
Somers' D with variable: studytime
Transformation: Harrell's c
Within strata defined by: agegp
Valid observations: 48
Symmetric 95% CI for Harrell's c
------------------------------------------------------------------------------
             |              Jackknife
   studytime |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        drug |   .7630597   .0398266    19.16   0.000      .685001    .8411184
       youth |   .5559701   .0607348     9.15   0.000     .4369321    .6750082
------------------------------------------------------------------------------
. lincom drug-youth
 ( 1)  drug - youth = 0
------------------------------------------------------------------------------
   studytime |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
```

```
         (1) |   .2070896   .0660029     3.14   0.002     .0777262    .3364529
------------------------------------------------------------------------------
```

This time, we see that youth is a less impressive predictor of survival *within age tertiles* (as the confidence interval for Harrell's $c$ contains 0.5), and is a less good predictor than treatment when predicting survival between subjects in the same age tertile. We can therefore conclude (in very strong terms) that treatment has an effect that is not *entirely* caused by confounding by age.

In this analysis, there is only one confounder. There are often many confounders in observational studies in real life, and this makes stratified analyses more difficult. However, a possible solution might be to define a propensity score, measuring proneness to allocation to a treatment and dependent on all the confounders, and to use `xtile` on the propensity score to define a propensity group variable, which can be used as the `wstrata()` option by `somersd`. The seminal paper on propensity scores is Rosenbaum and Rubin (1983), but a good place to start a literature search now might be Imai and van Dyk (2004).

Harrell's $c$ is usually used to compare the predictive power of multivariate predictive scores such as log hazard ratios from Cox regression models, rather than the univariate predictors assessed here. The `estat` utility, used after Cox regression, can calculate Harrell's $c$ and the corresponding Somers' $D$ for the log hazard ratio with respect to survival. It is demonstrated, using the dataset used here, in [ST] **stcox postestimation**. `estat` sometimes calculates values for Harrell's $c$ and Somers' $D$ different from those calculated by `somersd`. This is because `estat` assumes that, if two subjects have equal lifetimes and one ends in death and the other ends in censorship, then the second has survived the first, whereas `somersd` assumes that neither has survived the other, based on the csign$(.,.,.,.)$ function of (2). If the survival times are expressed in integral numbers of units (such as days), then such ties can be broken by adding a fraction of a unit to the censored lifetimes and not to the uncensored lifetimes. This will cause `somersd` to give the same values as `estat`.

It must be stressed that confidence limits for the Harrell's $c$ of a Cox regression score with respect to survival should be treated with great caution if they are calculated from the same dataset in which the Cox model parameters were fitted. Harrell *et al.* (1982) and Harrell *et al.* (1996) stress this, and propose some solutions, using a sub–sampling method similar to the bootstrap. When these methods are used on large datasets, the search tree algorithm of Newson (2006a), used in default by `somersd`, will require appreciably less time than the trivial quadratic–time algorithms used by many other programs (and by `somersd` if the user specifies `notree`).

The Wilcoxon–Breslow–Gehan test was introduced by Breslow (1970) and Gehan (1965), and is demonstrated using Stata in [ST] **sts test**. It tests the hypothesis of a zero value of the Somers' $D$ of survival (as the $Y$–variable) with respect to membership of a particular group (as the $X$–variable). Using `somersd`, we can improve on this by defining a confidence interval for this Somers' $D$ parameter:

```
. sts test drug, wilcoxon
         failure _d:  died
   analysis time _t:  studytime
Wilcoxon (Breslow) test for equality of survivor functions
      |   Events         Events       Sum of
drug  |  observed       expected       ranks
------+--------------------------------------
0     |        19           7.25         385
1     |        12          23.75        -385
------+--------------------------------------
Total |        31          31.00           0
            chi2(1) =      22.61
            Pr>chi2 =     0.0000
. somersd drug studytime, tr(z) cenind(0 censind)
Somers' D with variable: drug
Transformation: Fisher's z
Valid observations: 48
Symmetric 95% CI for transformed Somers' D
------------------------------------------------------------------------------
             |              Jackknife
        drug |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   studytime |   .8297787   .1935732     4.29   0.000     .4503821    1.209175
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Somers' D
                Somers_D     Minimum     Maximum
   studytime   .68035714   .42221306   .83643191
```

We see (from the Wilcoxon test) that the treated group has fewer deaths, and the placebo group has more deaths, than we would expect by chance, assuming population survival distributions to be the same in the two groups. We also see (from the confidence interval for the untransformed Somers' $D$) that, if we sample a subject at random from each of the two subpopulations (treated and placebo), then the event that the treated subject survives the placebo subject is 42% to 84% more probable than the event that the placebo subject survives the treated subject. Again, we can stratify the analysis by age tertile:

```
. somersd drug studytime, tr(z) cenind(0 censind) wstrata(agegp)
Somers' D with variable: drug
Transformation: Fisher's z
Within strata defined by: agegp
Valid observations: 48
Symmetric 95% CI for transformed Somers' D
------------------------------------------------------------------------------
             |              Jackknife
        drug |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   studytime |   .9729551   .2404965     4.05   0.000     .5015905     1.44432
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Somers' D
                Somers_D     Minimum     Maximum
   studytime         .75   .46336709   .89456394
```

We see that, if we sample a subject at random *from the same age tertile* in each of the two treatment groups (treated and placebo), then it is 46% to 89% more likely that the treated subject survives the untreated subject than *vice versa*.

Note that Harrell's $c$ and the Wilcoxon–Breslow–Gehan test are based on Somers $D$ parameters "in different directions", in that the survival variable is the first variable in the list when calculating Harrell's $c$ and the second variable in the list when calculating the Wilcoxon–Breslow–Gehan Somers' $D$. The two methods are complementary. The Harrell method is more useful for the "purely scientific" inference of establishing that a predictor has some predictive power that is not entirely caused by a confounder. The Wilcoxon–Breslow–Gehan method is more useful for estimating the size of the "causal effect" on survivorship of choosing one treatment.instead of another.

### 3.4 Gini coefficients in the `womenwage` data

The Gini coefficient of inequality is used by econometricians as a measure of the inequality of a distribution of incomes (or wealth) in a population, on a scale from zero (when everybody has an equal share) to one (when one person has everything). It is traditionally understood by reference to the Lorenz curve, which is the set of $(X, Y)$–points on the unit square such that the richest $100Y$ percent of the population have $100X$ percent of the income (or wealth). The Gini coefficient is equal to the difference between the area above the Lorenz curve and the area below the Lorenz curve. Note that Gini also invented several other coefficients, which are also referred to in various contexts as "the Gini coefficient", including Gini's mean difference, defined in Chapter 5 of Serfling (1980). These are summarized in Goodman and Kruskal (1959).

The Gini coefficient of inequality is a special case of Somers' $D$. To see this, imagine that two lotteries are organized in a population, and that, in the first lottery, each member of the population has one ticket, whereas, in the second lottery, each individual buys a number of tickets proportional to that individual's income. The first lottery is equivalent to sampling uniformly from the $Y$–axis of the Lorenz plot, whereas the second lottery is equivalent to sampling uniformly from the $X$–axis of the Lorenz plot. The region above the Lorenz curve corresponds to the event that the second lottery winner is a higher earner than the first, whereas the region below the Lorenz curve corresponds to the event that the first lottery winner is a higher earner than the second. Therefore, the Gini coefficient is a clustered Somers' $D$, where the clusters are individuals in the population, the observations are combinations of individual and lottery (first or second), the $Y$–variate is income, the $X$–variate is lottery sequence (1 or 2), and the importance weights are equal for all individuals in the first lottery and proportional to income for all individuals in the second lottery. The Lorenz curve is similar to the ROC curve (see [R] `roc`, Hanley and McNeil (1982), or Newson (2002)), except that the ROC curve demonstrates the outcome of sampling equiprobably from two sub–populations, whereas the Lorenz curve demonstrates the outcome of sampling twice from the same population, with different probability weights.

We can illustrate this principle using the `womenwage` dataset, distributed by Stata Press and used in [R] `intreg`. We first calculate the Gini index (and a few other inequality indices), using the program `ineqdeco` (Jenkins, 1999). We then preserve the data, and use the `expgen` package (an extended version of `expand` downloadable from SSC) to replace each observation in the original dataset (containing one observation per woman) with 2 observations (one per woman per lottery). The new dataset is indexed by the variables `womanid` (denoting sequence number of the woman) and `lotseq` (denoting sequence number of the lottery). We create an importance variable `pwt`, containing probability weights equal for all women in the first lottery and equal to a woman's wage (to the nearest kilodollar) in the second lottery. We then use `somersd` twice, first without any tranformation, and second with the normalizing and/or variance–stabilizing $z$–transformation, before restoring the old dataset:

```
. use http://www.stata-press.com/data/r9/womenwage, clear
(Wages of women)
. ineqdeco wage
Percentile ratios
----------------------------------------------------------
  All obs |    p90/p10     p90/p50     p10/p50     p75/p25
----------+-----------------------------------------------
          |      3.222       1.933       0.600       1.909
----------------------------------------------------------
Generalized Entropy indices GE(a), where a = income difference
 sensitivity parameter, and Gini coefficient
----------------------------------------------------------------------
  All obs |     GE(-1)       GE(0)       GE(1)       GE(2)        Gini
----------+-----------------------------------------------------------
          |    0.14595     0.12947     0.13383     0.16022     0.27984
----------------------------------------------------------------------
Atkinson indices, A(e), where e > 0 is the inequality aversion parameter
----------------------------------------------
  All obs |     A(0.5)        A(1)        A(2)
----------+-----------------------------------
          |    0.06358     0.12144     0.22594
----------------------------------------------
. preserve
. expgen =2, oldseq(womanid) copyseq(lotseq)
. lab var lotseq "Lottery sequence number"
. gene pwt = (lotseq==1) + wage*(lotseq==2)
. lab var pwt "Probability weight"
. somersd lotseq wage [pwei=pwt], cluster(womanid) funtype(vonmises)
Von Mises Somers' D with variable: lotseq
Transformation: Untransformed
Valid observations: 976
Number of clusters: 488
Symmetric 95% CI
                              (Std. Err. adjusted for 488 clusters in womanid)
------------------------------------------------------------------------------
             |              Jackknife
      lotseq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        wage |   .2798363   .0105714    26.47   0.000     .2591168    .3005558
------------------------------------------------------------------------------
. somersd lotseq wage [pwei=pwt], cluster(womanid) funtype(vonmises) tr(z)
Von Mises Somers' D with variable: lotseq
Transformation: Fisher's z
Valid observations: 976
Number of clusters: 488
Symmetric 95% CI for transformed Somers' D
                              (Std. Err. adjusted for 488 clusters in womanid)
------------------------------------------------------------------------------
             |              Jackknife
      lotseq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        wage |   .2875044   .0114695    25.07   0.000     .2650246    .3099843
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Somers' D
                Somers_D     Minimum     Maximum
        wage   .27983629   .25898919   .30042278
. restore
```

We see that, if the women in this dataset organized two lotteries amongst themselves, and each woman bought one ticket in the first lottery and a number of tickets worth a constant fraction of her wages in the second lottery, then the second lottery winner would be 27.98% more likely than the first lottery winner to be the higher earner of the two. (The option `funtype(vonmises)` is necessary because of the small but nonzero probability that the same woman wins both lotteries.) And, if the same lotteries were organized in the population from which these women were sampled, then the difference would be from 25.91% to 30.06% (according to the untransformed confidence interval), or 25.90% to 30.04% (according to the $z$–transformed confidence interval). In general, we expect normalizing and/or variance–stabilizing transformations to be more important in populations where the population Gini coefficient is higher, because then the sampling distribution of the *sample* Gini coefficient will be more skewed. This principle is discussed in Daniels and Kendall (1947) and Edwardes (1995).

Income distribution indices are often calculated on datasets in which each observation represents an income group, rather than an individual. We can create such a dataset using `contract` on the original `womanwage` data to produce a new dataset, with 1 observation per wage group. If we do this, then we can repeat our analysis to demonstrate the `cfweight()` option of `somersd`. We first use `expgen` to expand the dataset with 1 observation per wage group to a dataset with 1 observation per wage group per lottery, and create the weight variable as before. This new dataset contains a cluster of 2 variables per wage group, and each of these clusters represents a number of duplicate clusters equal to the value of the variable `_freq` created by `contract`. When calling `somersd`, we use `_freq` as a `cfweight()` variable, and produce the same results as before:

```
. preserve
. contract wage
. expgen =2, oldseq(wagegp) copyseq(lotseq)
. lab var wagegp "Wage group"
. lab var lotseq "Lottery sequence number"
. gene pwt = (lotseq==1) + wage*(lotseq==2)
. lab var pwt "Probability weight"
. describe
Contains data from http://www.stata-press.com/data/r9/womenwage.dta
  obs:           102                          Wages of women
 vars:             5                          3 Mar 2005 18:14
-------------------------------------------------------------------------------
              storage  display    value
variable name   type   format     label      variable label
-------------------------------------------------------------------------------
wage            float  %9.0g                 wages in 1000s of dollars
_freq           byte   %12.0g                Frequency
wagegp          byte   %12.0g                Wage group
lotseq          byte   %12.0g                Lottery sequence number
pwt             float  %9.0g                 Probability weight
-------------------------------------------------------------------------------
Sorted by: wagegp  lotseq
     Note: Dataset has changed since last saved.
. somersd lotseq wage [pwei=pwt], cluster(wagegp) cfweight(_freq) funtype(vonmi
> ses)
Von Mises Somers' D with variable: lotseq
Transformation: Untransformed
Valid observations: 976
Number of clusters: 488
Symmetric 95% CI
                               (Std. Err. adjusted for 488 clusters in wagegp)
------------------------------------------------------------------------------
             |              Jackknife
      lotseq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        wage |   .2798363   .0105714    26.47   0.000     .2591168    .3005558
------------------------------------------------------------------------------
. somersd lotseq wage [pwei=pwt], cluster(wagegp) cfweight(_freq) funtype(vonmi
> ses) tr(z)
Von Mises Somers' D with variable: lotseq
Transformation: Fisher's z
Valid observations: 976
Number of clusters: 488
Symmetric 95% CI for transformed Somers' D
                               (Std. Err. adjusted for 488 clusters in wagegp)
------------------------------------------------------------------------------
             |              Jackknife
      lotseq |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        wage |   .2875044   .0114695    25.07   0.000     .2650246    .3099843
------------------------------------------------------------------------------
Asymmetric 95% CI for untransformed Somers' D
                Somers_D     Minimum     Maximum
        wage   .27983629   .25898919   .30042278
. restore
```

## 4 Historical note

This document is a post–publication update of an article which appeared in the Stata Technical Bulletin (STB) as Newson (2000a). The somersd package was later revised in Newson (2000b), Newson (2000c), Newson (2000d), Newson (2001a), Newson (2001b) and Newson (2006b). An important upgrade (Newson, 2000d) was the addition to the somersd package of the program cendif, which calculates robust confidence intervals for Hodges–Lehmann median differences, other percentile differences, and percentile ratios. A post–publication update of that STB article is distributed with this document as part of the documentation to the somersd package. After 2001, STB was replaced by The Stata Journal (SJ), and most subsequent updates only appeared on SSC and on Roger Newson's homepage at *http://www.rogernewsonresources.org.uk/*, which was accessible from within net–aware Stata as of 16 April 2020. However, Newson (2002) gives a comprehensive review of Somers' $D$, Kendall's $\tau_a$, median differences, and their estimation in Stata using the somersd package. Also, Newson (2006a), Newson (2006b) and Newson (2006c) describe the upgrade of somersd to Version 9 of Stata, which greatly extended the range of versions of Somers' $D$ and Kendall's $\tau_a$ available, and used the Mata compilable matrix language for increased speed with large samples.

## 5 Acknowledgements

I would like to thank William Gould of Stata Corporation for suggesting the predict program somers_p, and Nicholas J. Cox of Durham University, UK, for some very helpful discussions on Somers' $D$ and Kendall's $\tau_a$.

## 6 References

Arvesen, J. N. 1969. Jackknifing U–statistics. *The Annals of Mathematical Statistics* 40: 2076–2100.

Breslow, N. E. 1970. A generalized Kruskal–Wallis test for comparing $k$ samples subject to unequal patterns of censorship. *Biometrika* 57: 579–594.

Daniels, H. E. and Kendall, M. G. 1947. The significance of rank correlation where parental correlation exists. *Biometrika* 34: 197–208.

Edwardes, M. D. deB. 1995. A confidence interval for Pr($X < Y$) - Pr($X > Y$) estimated from simple cluster samples. *Biometrics* 51: 571–578.

Fisher, R. A. 1921. On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron* 1(4): 3–32.

Gayen, A. K. 1951. The frequency distribution of the product–moment correlation coefficient in random samples of any size drawn from non–normal universes. *Biometrika* 38: 219–247.

Gehan, E. A. 1965. A generalized Wilcoxon test for comparing arbitrarily single–censored samples. *Biometrika* 52; 203–223.

Goodman, L. A. and W. H. Kruskal. 1959. Measures of association for cross classifications. II: Further discussion and references. *Journal of the American Statistical Association* 54(285): 123–163.

Hanley, J. A. and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143: 29–36.

Harrell, F. E., R. M. Califf, D. B. Pryor, K. L. Lee and R. A. Rosati. 1982. Evaluating the yield of medical tests. *Journal of the American Medical Association* 247(18): 2543–2546.

Harrell, F. E., K. L. Lee and D. B. Mark. 1996. Multivariate prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in Medicine* 15: 361–387.

Hoeffding, W. 1948. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics* 19: 293–325.

Imai K. and D. A. van Dyk. Causal inference with general treatment regimes: generalizing the propensity score. 2004. *Journal of the American Statistical Association* 99(467): 854–866.

Jenkins, S. P. 1999. Analysis of income distributions. *Stata Technical Bulletin* 48: 4–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 243–260.

Kendall, M. G. 1949. Rank and product–moment correlation. *Biometrika* 36: 177–193.

Kendall, M. G. and J. D. Gibbons. 1990. *Rank Correlation Methods. 5th edition.* New York: Oxford University Press.

Newson, R. 2000a. snp15: somersd – Confidence intervals for nonparametric statistics and their differences. *Stata Technical Bulletin* 55: 47–55. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 312–322.

Newson, R. 2000b. snp15.1: Update to somersd. *Stata Technical Bulletin* 57: 35. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 322–323.

Newson, R. 2000c. snp15.2: Update to somersd. *Stata Technical Bulletin* 58: 30. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, p. 323.

Newson, R. 2000d. snp16: Robust confidence intervals for median and other percentile differences between groups. *Stata Technical Bulletin* 58: 30–35. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 324–331.

Newson, R. 2001a. snp15.3: Update to somersd. *Stata Technical Bulletin* 61: 22. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, p. 33$x$.

Newson, R. 2001b. snp16.1: Update to cendif. *Stata Technical Bulletin* 61: 22. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, p. 33$x$.

Newson, R. 2002. Parameters behind "nonparametric" statistics: Kendall's tau, Somers' $D$ and median differences. *Stata Journal* 2: 45–64. A pre–publication draft can be downloaded from Roger Newson's website at http://www.rogernewsonresources.org.uk/

as of 16 April 2020, using the `net` command in Stata.

Newson, R. 2006a. Efficient calculation of jackknife confidence intervals for rank statistics. *Journal of Statistical Software* 15(1): 1–10. Downloadable from `http://www.jstatsoft.org/` as of 16 April 2020.

Newson, R. 2006b. Confidence intervals for rank statistics: Somers' $D$ and extensions. *Stata Journal* 6(3): 309–334. A pre-publication draft can be downloaded from Roger Newson's website at `http://www.rogernewsonresources.org.uk/` as of 16 April 2020, using the `net` command in Stata.

Newson, R. 2006c. Confidence intervals for rank statistics: Percentile slopes, differences, and ratios. *Stata Journal* 6(4): 497–520. A pre-publication draft can be downloaded from Roger Newson's website at `http://www.rogernewsonresources.org.uk/` as of 16 April 2020, using the `net` command in Stata.

Rosenbaum P. R. and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70(1): 41–55.

Serfling, R. J. 1980. Approximation theorems of mathematical statistics. New York: John Wiley & Sons.

Somers, R. H. 1962. A new asymmetric measure of association for ordinal variables. *American Sociological Review* 27: 799–811.