```
        sxy += xt*yt;
    }
    *r=sxy/(sqrt(sxx*syy)+TINY);
    *z=0.5*log((1.0+(*r)+TINY)/(1.0-(*r)+TINY));      Fisher's z transformation.
    df=n-2;
    t=(*r)*sqrt(df/((1.0-(*r)+TINY)*(1.0+(*r)+TINY)));      Equation (14.5.5).
    *prob=betai(0.5*df,0.5,df/(df+t*t));                    Student's t probability.
/*    *prob=erfcc(fabs((*z)*sqrt(n-1.0))/1.4142136)    */
For large n, this easier computation of prob, using the short routine erfcc, would give approx-
imately the same value.
}
```

CITED REFERENCES AND FURTHER READING:

Dunn, O.J., and Clark, V.A. 1974, *Applied Statistics: Analysis of Variance and Regression* (New York: Wiley).

Hoel, P.G. 1971, *Introduction to Mathematical Statistics*, 4th ed. (New York: Wiley), Chapter 7.

von Mises, R. 1964, *Mathematical Theory of Probability and Statistics* (New York: Academic Press), Chapters IX(A) and IX(B).

Korn, G.A., and Korn, T.M. 1968, *Mathematical Handbook for Scientists and Engineers*, 2nd ed. (New York: McGraw-Hill), §19.7.

Norusis, M.J. 1982, *SPSS Introductory Guide: Basic Statistics and Operations*; and 1985, *SPSS-X Advanced Statistics Guide* (New York: McGraw-Hill).

14.6 Nonparametric or Rank Correlation

It is precisely the uncertainty in interpreting the significance of the linear correlation coefficient r that leads us to the important concepts of *nonparametric* or *rank correlation*. As before, we are given N pairs of measurements (x_i, y_i). Before, difficulties arose because we did not necessarily know the probability distribution function from which the x_i's or y_i's were drawn.

The key concept of nonparametric correlation is this: If we replace the value of each x_i by the value of its *rank* among all the other x_i's in the sample, that is, $1, 2, 3, \dots, N$, then the resulting list of numbers will be drawn from a perfectly known distribution function, namely uniformly from the integers between 1 and N, inclusive. Better than uniformly, in fact, since if the x_i's are all distinct, then each integer will occur precisely once. If some of the x_i's have identical values, it is conventional to assign to all these "ties" the mean of the ranks that they would have had if their values had been slightly different. This *midrank* will sometimes be an integer, sometimes a half-integer. In all cases the sum of all assigned ranks will be the same as the sum of the integers from 1 to N, namely $\frac{1}{2}N(N+1)$.

Of course we do exactly the same procedure for the y_i's, replacing each value by its rank among the other y_i's in the sample.

Now we are free to invent statistics for detecting correlation between uniform sets of integers between 1 and N, keeping in mind the possibility of ties in the ranks. There is, of course, some loss of information in replacing the original numbers by ranks. We could construct some rather artificial examples where a correlation could be detected parametrically (e.g., in the linear correlation coefficient r), but could not be detected nonparametrically. Such examples are very rare in real life, however, and the slight loss of information in ranking is a small price to pay for a very major advantage: When a correlation is demonstrated to be present nonparametrically, then it is really there! (That is, to a certainty level that depends on the significance chosen.) Nonparametric correlation is more robust than linear correlation, more resistant to unplanned defects in the data, in the same sort of sense that the median is more robust than the mean. For more on the concept of robustness, see §15.7.

As always in statistics, some particular choices of a statistic have already been invented for us and consecrated, if not beatified, by popular use. We will discuss two, the *Spearman rank-order correlation coefficient* (r_s), and *Kendall's tau* (τ).

Spearman Rank-Order Correlation Coefficient

Let R_i be the rank of x_i among the other x's, S_i be the rank of y_i among the other y's, ties being assigned the appropriate midrank as described above. Then the rank-order correlation coefficient is defined to be the linear correlation coefficient of the ranks, namely,

$$r_s = \frac{\sum_i (R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_i (R_i - \overline{R})^2}\sqrt{\sum_i (S_i - \overline{S})^2}} \tag{14.6.1}$$

The significance of a nonzero value of r_s is tested by computing

$$t = r_s \sqrt{\frac{N-2}{1-r_s^2}} \tag{14.6.2}$$

which is distributed approximately as Student's distribution with $N-2$ degrees of freedom. A key point is that this approximation does not depend on the original distribution of the x's and y's; it is always the same approximation, and always pretty good.

It turns out that r_s is closely related to another conventional measure of nonparametric correlation, the so-called *sum squared difference of ranks*, defined as

$$D = \sum_{i=1}^{N} (R_i - S_i)^2 \tag{14.6.3}$$

(This D is sometimes denoted D^{**}, where the asterisks are used to indicate that ties are treated by midranking.)

When there are no ties in the data, then the exact relation between D and r_s is

$$r_s = 1 - \frac{6D}{N^3 - N} \tag{14.6.4}$$

When there are ties, then the exact relation is slightly more complicated: Let f_k be the number of ties in the kth group of ties among the R_i's, and let g_m be the number of ties in the mth group of ties among the S_i's. Then it turns out that

$$r_s = \frac{1 - \dfrac{6}{N^3 - N}\left[D + \frac{1}{12}\sum_k (f_k^3 - f_k) + \frac{1}{12}\sum_m (g_m^3 - g_m)\right]}{\left[1 - \dfrac{\sum_k (f_k^3 - f_k)}{N^3 - N}\right]^{1/2}\left[1 - \dfrac{\sum_m (g_m^3 - g_m)}{N^3 - N}\right]^{1/2}} \tag{14.6.5}$$

holds exactly. Notice that if all the f_k's and all the g_m's are equal to one, meaning that there are no ties, then equation (14.6.5) reduces to equation (14.6.4).

In (14.6.2) we gave a t-statistic that tests the significance of a nonzero r_s. It is also possible to test the significance of D directly. The expectation value of D in the null hypothesis of uncorrelated data sets is

$$\overline{D} = \frac{1}{6}(N^3 - N) - \frac{1}{12}\sum_k (f_k^3 - f_k) - \frac{1}{12}\sum_m (g_m^3 - g_m) \tag{14.6.6}$$

its variance is

$$\begin{aligned}\text{Var}(D) = {} & \frac{(N-1)N^2(N+1)^2}{36} \\ & \times \left[1 - \frac{\sum_k (f_k^3 - f_k)}{N^3 - N}\right]\left[1 - \frac{\sum_m (g_m^3 - g_m)}{N^3 - N}\right]\end{aligned} \tag{14.6.7}$$

and it is approximately normally distributed, so that the significance level is a complementary error function (cf. equation 14.5.2). Of course, (14.6.2) and (14.6.7) are not independent tests, but simply variants of the same test. In the program that follows, we calculate both the significance level obtained by using (14.6.2) and the significance level obtained by using (14.6.7); their discrepancy will give you an idea of how good the approximations are. You will also notice that we break off the task of assigning ranks (including tied midranks) into a separate function, `crank`.

```
#include <math.h>
#include "nrutil.h"

void spear(float data1[], float data2[], unsigned long n, float *d, float *zd,
    float *probd, float *rs, float *probrs)
```

Given two data arrays, `data1[1..n]` and `data2[1..n]`, this routine returns their sum-squared difference of ranks as D, the number of standard deviations by which D deviates from its null-hypothesis expected value as `zd`, the two-sided significance level of this deviation as `probd`, Spearman's rank correlation r_s as `rs`, and the two-sided significance level of its deviation from zero as `probrs`. The external routines `crank` (below) and `sort2` (§8.2) are used. A small value of either `probd` or `probrs` indicates a significant correlation (`rs` positive) or anticorrelation (`rs` negative).

```
{
    float betai(float a, float b, float x);
    void crank(unsigned long n, float w[], float *s);
    float erfcc(float x);
    void sort2(unsigned long n, float arr[], float brr[]);
    unsigned long j;
    float vard,t,sg,sf,fac,en3n,en,df,aved,*wksp1,*wksp2;

    wksp1=vector(1,n);
    wksp2=vector(1,n);
    for (j=1;j<=n;j++) {
        wksp1[j]=data1[j];
        wksp2[j]=data2[j];
    }
    sort2(n,wksp1,wksp2);          Sort each of the data arrays, and convert the entries to
    crank(n,wksp1,&sf);            ranks. The values sf and sg return the sums ∑(f_k^3−f_k)
    sort2(n,wksp2,wksp1);          and ∑(g_m^3 − g_m), respectively.
    crank(n,wksp2,&sg);
    *d=0.0;
    for (j=1;j<=n;j++)             Sum the squared difference of ranks.
        *d += SQR(wksp1[j]-wksp2[j]);
    en=n;
    en3n=en*en*en-en;
    aved=en3n/6.0-(sf+sg)/12.0;                           Expectation value of D,
    fac=(1.0-sf/en3n)*(1.0-sg/en3n);
    vard=((en-1.0)*en*en*SQR(en+1.0)/36.0)*fac;           and variance of D give
    *zd=(*d-aved)/sqrt(vard);                             number of standard devia-
    *probd=erfcc(fabs(*zd)/1.4142136);                    tions and significance.
    *rs=(1.0-(6.0/en3n)*(*d+(sf+sg)/12.0))/sqrt(fac);     Rank correlation coefficient,
    fac=(*rs+1.0)*(1.0-(*rs));
    if (fac > 0.0) {
        t=(*rs)*sqrt((en-2.0)/fac);                       and its t value,
        df=en-2.0;
        *probrs=betai(0.5*df,0.5,df/(df+t*t));            give its significance.
    } else
        *probrs=0.0;
    free_vector(wksp2,1,n);
    free_vector(wksp1,1,n);
}
```

```
void crank(unsigned long n, float w[], float *s)
```

Given a sorted array `w[1..n]`, replaces the elements by their rank, including midranking of ties, and returns as `s` the sum of $f^3 - f$, where f is the number of elements in each tie.

```
{
    unsigned long j=1,ji,jt;
    float t,rank;

    *s=0.0;
    while (j < n) {
        if (w[j+1] != w[j]) {           Not a tie.
            w[j]=j;
            ++j;
        } else {                        A tie:
            for (jt=j+1;jt<=n && w[jt]==w[j];jt++);      How far does it go?
            rank=0.5*(j+jt-1);          This is the mean rank of the tie,
            for (ji=j;ji<=(jt-1);ji++) w[ji]=rank;       so enter it into all the tied
            t=jt-j;                                          entries,
            *s += t*t*t-t;              and update s.
            j=jt;
        }
    }
    if (j == n) w[n]=n;                 If the last element was not tied, this is its rank.
}
```

Kendall's Tau

Kendall's τ is even more nonparametric than Spearman's r_s or D. Instead of using the numerical difference of ranks, it uses only the relative ordering of ranks: higher in rank, lower in rank, or the same in rank. But in that case we don't even have to rank the data! Ranks will be higher, lower, or the same if and only if the values are larger, smaller, or equal, respectively. On balance, we prefer r_s as being the more straightforward nonparametric test, but both statistics are in general use. In fact, τ and r_s are very strongly correlated and, in most applications, are effectively the same test.

To define τ, we start with the N data points (x_i, y_i). Now consider all $\frac{1}{2}N(N-1)$ *pairs* of data points, where a data point cannot be paired with itself, and where the points in either order count as one pair. We call a pair *concordant* if the relative ordering of the ranks of the two x's (or for that matter the two x's themselves) is the same as the relative ordering of the ranks of the two y's (or for that matter the two y's themselves). We call a pair *discordant* if the relative ordering of the ranks of the two x's is opposite from the relative ordering of the ranks of the two y's. If there is a tie in either the ranks of the two x's or the ranks of the two y's, then we don't call the pair either concordant or discordant. If the tie is in the x's, we will call the pair an "extra y pair." If the tie is in the y's, we will call the pair an "extra x pair." If the tie is in both the x's and the y's, we don't call the pair anything at all. Are you still with us?

Kendall's τ is now the following simple combination of these various counts:

$$\tau = \frac{\text{concordant} - \text{discordant}}{\sqrt{\text{concordant} + \text{discordant} + \text{extra-}y}\,\sqrt{\text{concordant} + \text{discordant} + \text{extra-}x}} \tag{14.6.8}$$

You can easily convince yourself that this must lie between 1 and -1, and that it takes on the extreme values only for complete rank agreement or complete rank reversal, respectively.

More important, Kendall has worked out, from the combinatorics, the approximate distribution of τ in the null hypothesis of no association between x and y. In this case τ is approximately normally distributed, with zero expectation value and a variance of

$$\text{Var}(\tau) = \frac{4N + 10}{9N(N-1)} \tag{14.6.9}$$

The following program proceeds according to the above description, and therefore loops over all pairs of data points. Beware: This is an $O(N^2)$ algorithm, unlike the algorithm for r_s, whose dominant sort operations are of order $N \log N$. If you are routinely computing Kendall's τ for data sets of more than a few thousand points, you may be in for some serious computing. If, however, you are willing to bin your data into a moderate number of bins, then read on.

```
#include <math.h>

void kendl1(float data1[], float data2[], unsigned long n, float *tau,
    float *z, float *prob)
```

Given data arrays `data1[1..n]` and `data2[1..n]`, this program returns Kendall's τ as `tau`, its number of standard deviations from zero as `z`, and its two-sided significance level as `prob`. Small values of `prob` indicate a significant correlation (`tau` positive) or anticorrelation (`tau` negative).

```
{
    float erfcc(float x);
    unsigned long n2=0,n1=0,k,j;
    long is=0;
    float svar,aa,a2,a1;

    for (j=1;j<n;j++) {                          Loop over first member of pair,
        for (k=(j+1);k<=n;k++) {                 and second member.
            a1=data1[j]-data1[k];
            a2=data2[j]-data2[k];
            aa=a1*a2;
            if (aa) {                            Neither array has a tie.
                ++n1;
                ++n2;
                aa > 0.0 ? ++is : --is;
            } else {                                One or both arrays have ties.
                if (a1) ++n1;                       An "extra x" event.
                if (a2) ++n2;                       An "extra y" event.
            }
        }
    }
    *tau=is/(sqrt((double) n1)*sqrt((double) n2));  Equation (14.6.8).
    svar=(4.0*n+10.0)/(9.0*n*(n-1.0));              Equation (14.6.9).
    *z=(*tau)/sqrt(svar);
    *prob=erfcc(fabs(*z)/1.4142136);                Significance.
}
```

Sometimes it happens that there are only a few possible values each for x and y. In that case, the data can be recorded as a contingency table (see §14.4) that gives the number of data points for each contingency of x and y.

Spearman's rank-order correlation coefficient is not a very natural statistic under these circumstances, since it assigns to each x and y bin a not-very-meaningful midrank value and then totals up vast numbers of identical rank differences. Kendall's tau, on the other hand, with its simple counting, remains quite natural. Furthermore, its $O(N^2)$ algorithm is no longer a problem, since we can arrange for it to loop over pairs of contingency table entries (each containing many data points) instead of over pairs of data points. This is implemented in the program that follows.

Note that Kendall's tau can be applied only to contingency tables where both variables are *ordinal*, i.e., well-ordered, and that it looks specifically for monotonic correlations, not for arbitrary associations. These two properties make it less general than the methods of §14.4, which applied to *nominal*, i.e., unordered, variables and arbitrary associations.

Comparing `kendl1` above with `kendl2` below, you will see that we have "floated" a number of variables. This is because the number of events in a contingency table might be sufficiently large as to cause overflows in some of the integer arithmetic, while the number of individual data points in a list could not possibly be that large [for an $O(N^2)$ routine!].

```
#include <math.h>

void kendl2(float **tab, int i, int j, float *tau, float *z, float *prob)
```

Given a two-dimensional table `tab[1..i][1..j]`, such that `tab[k][l]` contains the number of events falling in bin `k` of one variable and bin `l` of another, this program returns Kendall's τ as `tau`, its number of standard deviations from zero as `z`, and its two-sided significance level as `prob`. Small values of `prob` indicate a significant correlation (`tau` positive) or anticorrelation (`tau` negative) between the two variables. Although `tab` is a `float` array, it will normally contain integral values.

```
{
    float erfcc(float x);
    long nn,mm,m2,m1,lj,li,l,kj,ki,k;
    float svar,s=0.0,points,pairs,en2=0.0,en1=0.0;

    nn=i*j;                             Total number of entries in contingency table.
    points=tab[i][j];
    for (k=0;k<=nn-2;k++) {             Loop over entries in table,
        ki=(k/j);                       decoding a row,
        kj=k-j*ki;                      and a column.
        points += tab[ki+1][kj+1];      Increment the total count of events.
        for (l=k+1;l<=nn-1;l++) {       Loop over other member of the pair,
            li=l/j;                    decoding its row
            lj=l-j*li;                 and column.
            mm=(m1=li-ki)*(m2=lj-kj);
            pairs=tab[ki+1][kj+1]*tab[li+1][lj+1];
            if (mm) {                  Not a tie.
                en1 += pairs;
                en2 += pairs;
                s += (mm > 0 ? pairs : -pairs);     Concordant, or discordant.
            } else {
                if (m1) en1 += pairs;
                if (m2) en2 += pairs;
            }
        }
    }
    *tau=s/sqrt(en1*en2);
    svar=(4.0*points+10.0)/(9.0*points*(points-1.0));
    *z=(*tau)/sqrt(svar);
    *prob=erfcc(fabs(*z)/1.4142136);
}
```

CITED REFERENCES AND FURTHER READING:

Lehmann, E.L. 1975, *Nonparametrics: Statistical Methods Based on Ranks* (San Francisco: Holden-Day).

Downie, N.M., and Heath, R.W. 1965, *Basic Statistical Methods*, 2nd ed. (New York: Harper & Row), pp. 206–209.

Norusis, M.J. 1982, *SPSS Introductory Guide: Basic Statistics and Operations*; and 1985, *SPSS-X Advanced Statistics Guide* (New York: McGraw-Hill).

14.7 Do Two-Dimensional Distributions Differ?

We here discuss a useful generalization of the K–S test (§14.3) to *two-dimensional* distributions. This generalization is due to Fasano and Franceschini [1], a variant on an earlier idea due to Peacock [2].

In a two-dimensional distribution, each data point is characterized by an (x, y) pair of values. An example near to our hearts is that each of the 19 neutrinos that were detected from Supernova 1987A is characterized by a time t_i and by an energy E_i (see [3]). We might wish to know whether these measured pairs (t_i, E_i), $i = 1 \dots 19$ are consistent with a theoretical model that predicts neutrino flux as a function of both time and energy — that is, a two-dimensional probability distribution in the (x, y) [here, (t, E)] plane. That would be a one-sample test. Or, given two sets of neutrino detections, from two comparable detectors, we might want to know whether they are compatible with each other, a two-sample test.

In the spirit of the tried-and-true, one-dimensional K–S test, we want to range over the (x, y) plane in search of some kind of maximum *cumulative* difference between two two-dimensional distributions. Unfortunately, cumulative probability distribution is not well-defined in more than one dimension! Peacock's insight was that a good surrogate is the *integrated probability in each of four natural quadrants* around a given point (x_i, y_i), namely the total probabilities (or fraction of data) in $(x > x_i, y > y_i)$, $(x < x_i, y > y_i)$, $(x < x_i, y < y_i)$, $(x > x_i, y < y_i)$. The two-dimensional K–S statistic D is now taken to be the maximum difference (ranging both over data points and over quadrants) of the corresponding integrated probabilities. When comparing two data sets, the value of D may depend on which data set is ranged over. In that case, define an effective D as the average