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 \( x_j \)'s in the sample, that is, 1, 2, 3, ..., \( N \), 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 \( y_j \)'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 beati®ed, by popular use. We will discuss two, the Spearman rank-order correlation coefficient \((r_s)\), and Kendall's tau \((\tau)\).

**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,

\[
r_s = \frac{\sum_i (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2 \sum_i (S_i - \bar{S})^2}}
\]

(14.6.1)

The significance of a nonzero value of \(r_s\) is tested by computing

\[
t = r_s \sqrt{\frac{N - 2}{1 - r_s^2}}
\]

(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

\[
D = \sum_{i=1}^{N} (R_i - S_i)^2
\]

(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

\[
r_s = 1 - \frac{6D}{N^3 - N}
\]

(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 \(k\)th group of ties among the \(R_i\)'s, and let \(g_m\) be the number of ties in the \(m\)th group of ties among the \(S_i\)'s. Then it turns out that

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

(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

\[
D = \frac{1}{6}(N^3 - N) - \frac{1}{12} \sum_k (f^3_k - f_k) - \frac{1}{12} \sum_m (g^3_m - g_m) \tag{14.6.6}
\]

its variance is

\[
\text{Var}(D) = \frac{(N-1)N^2(N+1)^2}{36} \times \left[ 1 - \frac{\sum_k (f^3_k - f_k)}{N^3 - N} \right] \left[ 1 - \frac{\sum_m (g^3_m - g_m)}{N^3 - N} \right] \tag{14.6.7}
\]

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, \( \text{crank} \).

```c
#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)
{
    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, 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);
    crank(n, wksp1, &sf);
    sort2(n, wksp2, wksp1);
    crank(n, wksp2, &sg);
    d = 0.0;
    for (j = 1; j <= n; j++)
        *d += SQR(wksp1[j] - wksp2[j]);
    ...
```
void crank(unsigned long n, float w[], float *s)
Given a sorted array w[1..n], replaces the elements by their rank, including mid-ranking 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?
      t=(jt-j)*t; // 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+=t; // entries,
      s += t*t; // and update s.
      s += t*jt*jt;
      j=jt;
    }
  }
  if (j == n) w[n]=n; // If the last element was not tied, this is its rank.
}

Kendall's Tau

Kendall's $\tau$ 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, $\tau$ and $r_s$ are very strongly correlated and, in most applications, are effectively the same test.

To define $\tau$, 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
14.6 Nonparametric or Rank Correlation

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 $\tau$ is now the following simple combination of these various counts:

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

(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 $\tau$ in the null hypothesis of no association between $x$ and $y$. In this case $\tau$ is approximately normally distributed, with zero expectation value and a variance of

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

(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 $r$, whose dominant sort operations are of order $N \log N$. If you are routinely computing Kendall's $\tau$ 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.

```c
#include <math.h>

void kendl1(float data1[], float data2[], unsigned long n, float *tau, float *z, float *prob)
{
    float erfcc(float x);
    unsigned long n2=0,n1=0,k,j;
    long is=0;
    float svar,aa,a1,a2;

    for (j=1;j<n;j++) {
        for (k=(j+1);k<=n;k++) {
            a1=data1[j]-data1[k];
            a2=data2[j]-data2[k];
            aa=a1*a2;
            if (aa) {
                ++n1;
                ++n2;
                if (a1*aa<0) {
                    ++is;
                }
            } else {
                if (a1) {
                    ++n1;
                } else {
                    ++n2;
                }
            }
            svar+=aa;           // svar = sum of squares of differences
        }
    }
    svar*=2;
    svar+=svar;
    svar/=(n1+n2);
    svar=sqrt(svar*n1*n2/(n1+n2));
    *tau=0.5+0.5*erfcc(svar);
    *z=svar;
    *prob=0.5*(1.0+erf(svar/sqrt(2.0)));
}
```

Given data arrays `data1[1..n]` and `data2[1..n]`, this program returns Kendall's $\tau$ 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).

(14.6.10)
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 kend1l above with kend12 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!].

```c
#include <math.h>

void kend12(float **tab, int i, int j, float *tau, float *z, float *prob)
{   
    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++)
    {   
        ki=(k/j);  
        Loop over entries in table,
        kj=k-j*ki;
        and a column.
        points += tab[kj][li+1];
        Increment the total count of events.
        for (l=k+1;l<=nn-1;l++)
        {   
            Loop over other member of the pair,

++n2;
    aa > 0.0 ? ++is : --is;
  } else {
    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 kend1l above with kend12 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!].

```
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 \((t_i, E_i),\ i = 1 \ldots 19\) 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 \((x,y)\) 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

```c
li=1/j; decoding its row
lj=1-j*li;
mm=(m1=li-ki)*(m2=lj-kj);
pairs=tab[xi+1][kj+1]*tab[li+1][lj+1];
if (mm) {
    en1 += pairs;
en2 += pairs;
s += (mm > 0 ? pairs : -pairs);
    Not a tie.
} 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: