Lecture 32
Statistical description of data
Modeling of data

- Nonparametric or rank correlation.
- Spearman rank order coefficient.
- Kendall's tau.
- Modeling of data.
- Least squares.
- Chi-square

We have discussed the uncertainty in interpreting the significance of the linear correlation coefficient r. This leads us to the important concept of *nonparametric or rank correlation.* As before we consider N pairs of measurements (x_i,y_i).

We then replace the value of each x_i for its rank among the other x_i's, that is, 1,2,3,…,N. The resulting list of numbers is therefore drawn from a perfectly known distribution, namely uniformly from the integers from 1 to N. If two x_i's happen to be the same, there are ways to deal with "ties", namely assign each number a rank that is the mean of the one they would have had if they had been slightly different. The sum of all assigned ranks is the sum of the integers from 1 to N, namely N(N+1)/2.

We can now construct statistics for detecting correlations between uniform sets of integers from 1 to N. There is some information lost in replacing numbers by ranks. One could conceive of distributions that have correlations that are missed by doing things parametrically, but of course, they would be extremely special and non-generic.

Let us consider two statistics that have been developed for this purpose, the Spearman rank-order correlation coefficient and Kendall's tau.

The Spearman rank-order correlation coefficient

Let R_i be the rank of x_i among the other x's and S_i the rank of y_i among the y's. The rank-order correlation coefficient is defined as the linear correlation coefficient of the ranks, namely,

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

The significance of a nonzero value of r_s is tested by computing, $t = r_s \sqrt{\frac{N-2}{1-r_s^2}}$

It turns out that r_s is closely related to another usual measure of nonparametric correlation, the so-called sum squared difference of ranks $D = \sum_{i=1}^{N} (R_i - S_i)^2$

And the relation between them is $r_s = 1 - \frac{6D}{N^3 - N}$

These formulas assume no ties, Numerical Recipes has the generalized ones that can handle them.

```
      SUBROUTINE crank(n,w,s)
      INTEGER n
      REAL s,w(n)
```
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.
```
      INTEGER j,ji,jt
      REAL rank,t
      s=0.
      j=1                              The next rank to be assigned.
1     if(j.lt.n)then                   "do while" structure.
        if(w(j+1).ne.w(j))then         Not a tie.
          w(j)=j
          j=j+1
        else                           A tie:
          do 11 jt=j+1,n               How far does it go?
            if(w(jt).ne.w(j))goto 2
          enddo 11
          jt=n+1                       If here, it goes all the way to the last element.
2         rank=0.5*(j+jt-1)            This is the mean rank of the tie,
          do 12 ji=j,jt-1              so enter it into all the tied entries,
            w(ji)=rank
          enddo 12
          t=jt-j
          s=s+t**3-t                   and update s.
          j=jt
        endif
      goto 1
```

```
      SUBROUTINE spear(data1,data2,n,wksp1,wksp2,d,zd,probd,rs,probrs)
      INTEGER n
      REAL d,probd,probrs,rs,zd,data1(n),data2(n),wksp1(n),wksp2(n)
C     USES betai,crank,erfcc,sort2
```

Given two data arrays, `data1(1:n)` and `data2(1:n)`, each of length `n`, and given two workspaces of the same length, 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 workspaces can be identical to the data arrays, but in that case the data small value of either `probd` or `probrs` indicates a significant correlation (`rs` positive) or anticorrelation (`rs` negative).

```
      INTEGER j
      REAL aved,df,en,en3n,fac,sf,sg,t,vard,betai,erfcc
      do 11 j=1,n
        wksp1(j)=data1(j)
        wksp2(j)=data2(j)
      enddo 11
      call sort2(n,wksp1,wksp2)
      call crank(n,wksp1,sf)
      call sort2(n,wksp2,wksp1)
      call crank(n,wksp2,sg)
      d=0.
      do 12 j=1,n
        d=d+(wksp1(j)-wksp2(j))**2
      enddo 12
      en=n
      en3n=en**3-en
      aved=en3n/6.-(sf+sg)/12.
      fac=(1.-sf/en3n)*(1.-sg/en3n)
      vard=((en-1.)*en**2*(en+1.)**2/36.)*fac
      zd=(d-aved)/sqrt(vard)
      probd=erfcc(abs(zd)/1.4142136)
      rs=(1.-(6./en3n)*(d+(sf+sg)/12.))/sqrt(fac)
      fac=(1.+rs)*(1.-rs)
      if(fac.gt.0.)then
        t=rs*sqrt((en-2.)/fac)
        df=en-2.
        probrs=betai(0.5*df,0.5,df/(df+t**2))
      else
        probrs=0.
      endif
      return
      END
```

Sort each of the data arrays, and convert the entries to ranks. The values `sf` and `sg` return the sums $\sum(f_k^3 - f_k)$ and $\sum(g_m^3 - g_m)$, respectively.

Sum the squared difference of ranks.

Expectation value of D,

and variance of D give number of standard deviations, and significance.

Rank correlation coefficient,

and its t value,

give its significance.

Kendall's tau

Kendall's tau is even more non-parametric. Instead of using the numerical difference of ranks it uses only the relative order of the ranks: higher, lower or equal! Which means we don't even have to rank the data. It turns out that for most applications τ and r_s are strongly correlated and either can be effectively used.

To define the tau, we start again with N pairs of data points (x_i,y_i)a nd consider the N(N-1) /2 pairings of data points where a data point cannot be paired with itself and where the points in either order count as a pair. We call a pair concordant is the relative order of the x's is the same as that of the y's and discordant if it is not. If there is a tie in the x's we call the pair extra-x and if there is a tie in the y's, extra-y. Then we compute,

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

This lies between 1 and -1 takes extreme values only on complete rank agreement or rank reversal. Kendall has worked out the approximate distribution of t in the hypothesis of no association between x and y. In that case t is normally distributed with zero expectation value and a variance

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

```
      SUBROUTINE kendl1(data1,data2,n,tau,z,prob)
      INTEGER n
      REAL prob,tau,z,data1(n),data2(n)
C     USES erfcc
```

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).

```
      INTEGER is,j,k,n1,n2
      REAL a1,a2,aa,var,erfcc
      n1=0                                This will be the argument of one square root in (14.6.8),
      n2=0                                and this the other.
      is=0                                This will be the numerator in (14.6.8).
      do 12 j=1,n-1                       Loop over first member of pair,
        do 11 k=j+1,n                     and second member.
          a1=data1(j)-data1(k)
          a2=data2(j)-data2(k)
          aa=a1*a2
          if(aa.ne.0.)then                Neither array has a tie.
            n1=n1+1
            n2=n2+1
            if(aa.gt.0.)then
              is=is+1
            else
              is=is-1
            endif
          else                            One or both arrays have ties.
            if(a1.ne.0.)n1=n1+1                 An "extra $x$" event.
            if(a2.ne.0.)n2=n2+1                 An "extra $y$" event.
          endif
        enddo 11
      enddo 12
      tau=float(is)/sqrt(float(n1)*float(n2))   Equation (14.6.8).
      var=(4.*n+10.)/(9.*n*(n-1.))              Equation (14.6.9).
      z=tau/sqrt(var)
      prob=erfcc(abs(z)/1.4142136)              Significance.
      return
      END
```

Modeling of data

Given a data set one may want to see how well it fits with a “model” or theory that depends on certain parameters. For instance a polynomial of a certain degree.

The basic approach is to choose or design a “figure of merit function” (merit function for short) that measures the agreement between the data and the model with a particular choice of parameters. The parameters of the model are then adjusted to achieve a minimum in the merit function, yielding best fit parameters. The adjustment is therefore a problem of minimization in many dimensions.

There are additional issues that go beyond just fitting the parameters. Data usually contain errors. We therefore would like to have a way to assess if the model is appropriate. That is we need a test of goodness-of-fit against some useful statistical standard. We also need to know the accuracy with which the parameters are being determined. In other words, we need a measure of the error in the values of the parameters.

Finally, in some cases the merit function has several minima and we may be interested in the global minimum.

Summarizing, a fitting procedure should:

(i) Provide values for the parameters.
(ii) Provide errors for the parameters.
(iii) Provide some statistical measure of goodness of fit.

If (iii) is lousy, then (i) and (ii) are probably worthless, no matter what are their values.

Unfortunately, most physicists stop at (i)!!!

Least squares as maximum likelihood estimator

Suppose that you are fitting N data points (x_i,y_i) i=1,…,N, with a model that has M adjustable parameters a_j, j=1,…,M. The model predicts a functional relationship between the measured dependent and independent variables,

$$y(x) = y(x; a_1,\ldots, a_M)$$

The first thing that comes to mind as a likelihood estimator is to minimize the sum of the squares of the discrepancies,

$$\text{minimize over } a_1,\ldots,a_M: \qquad \sum_{i=1}^{N}\left[y_i - y(x; a_1,\ldots,a_M)\right]^2$$

This looks intuitively obvious. The quantity that we are minimizing is a sort of "distance" in functional space for a finite number of data points. But is there a better foundation for this?

Suppose that each data point has a random measurement error that is independent and distributed as a normal (Gaussian) distribution around the "true" model y(x). Let us assume that the standard deviations of these normal distributions are the same for all data points. Then the probability of the data set is the product of the probabilities of each point,

$$P \propto \prod_{i=1}^{N} \left\{ \exp\left(-\frac{1}{2}\left[\frac{y_i - y(x_i)}{\sigma} \right]^2 \right) \right\} \Delta y$$

We need to introduce Δy because the variables are continuous and asking for a probability of two values coinciding yields zero, one needs to ask for agreement in an interval.

Maximizing the probability is equivalent to maximizing its log, that is maximizing,

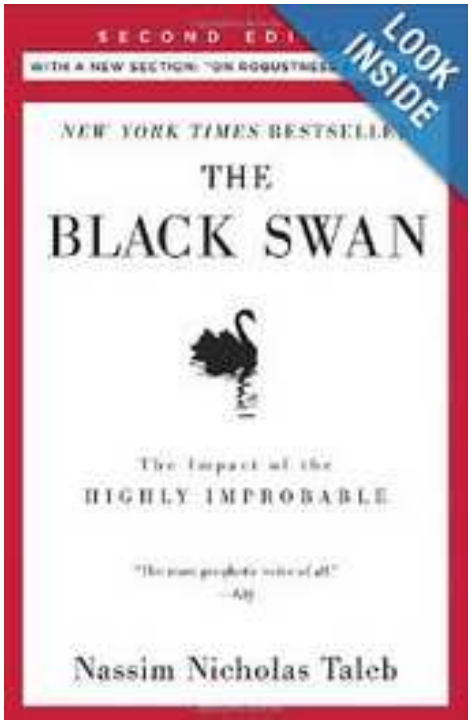
$$\left[\sum_{i=1}^{N} \frac{\left(y_i - y(x_i)\right)^2}{2\sigma^2} \right] - N \log(\Delta y)$$

And since N, Δy and s are constants, we get the same minimization as before.

This derivation teaches us several important things about least squares: it is a maximum likelihood estimator if the measurement errors are independent and normally distributed with constant standard deviations.

Notice that we made no assumption about the model. We will soon relax the requirement of constant standard deviations to yield “weighted least squares fitting”.

We need to ponder the heavy assumption of normal distribution. Because of the central limit theorem that states that the sum of a very large number of very small random deviations converges to the normal distribution, it is usually assumed that errors are normally distributed. We are taught that data are concentrated within one-sigma 68 percent of the time, +/-2 sigma 95 percent of the time, etc. This would suggest that an error of 20 sigma occurs only once ever 10^{88}. We all know that errors occur more frequently than that.



Sometimes the non-Gaussianity is easy to spot. For instance when measurements are obtained by counting events, they are usually distributed in Poisson distribution.
The latter tends to a Gaussian when the number of points is large, but the convergence is non uniform when measured in fractional accuracy. The more you are out in the "tail" the larger the number of points needed to have convergence: the Gaussian distribution invariably predicts that "tail" events occur less frequently than in the Poisson distribution.

Sometimes it is not so easy. Experimental points can diverge wildly due to external factors (someone kicked the equipment when that particular point was acquired).
Points like those are known as outliers. Their probability in a Gaussian distribution is so small that the maximum likelihood estimator distorts the whole curve to try to bring the outliers into line. We will discuss robust methods that can handle non Gaussian distributions later on.

Finally, the errors we are considering are statistical errors, the kind that will average out if one considers lots of measurements. One can also have systematic effects. For instance, the calibration of a measurement device may be temperature dependent.
If all the measurements are taken at the wrong temperature, no statistical massaging will correct for the unrecognized systematic error.

Chi-square fitting

If each data point (x_i,y_i) has its own known standard deviation σ_i, the formulas we considered change a bit. The maximum likelihood estimators is obtained by minimizing the quantity known as the Chi squared,

$$\chi^2 \equiv \sum_{i=1}^{N} \left(\frac{y_i - y(x_i; a_1, \ldots, a_M)}{\sigma_i} \right)$$

Taking the derivative yields,

$$0 = \sum_{i=1}^{N} \left(\frac{y_i - y(x_i)}{\sigma_i^2} \right) \frac{\partial y(x_i; a_1, \ldots, a_M)}{\partial a_k} \qquad k = 1, \ldots, M$$

And this is a set of M non-linear equations for the M unknowns.

The probability that chi squared exceeds a given value is given by the chi-square distribution, related to the incomplete Gamma function,

$$P(\chi^2 \mid N - M) = \frac{1}{\Gamma(\chi)} \int_0^{N-M} e^{-t} t^{\chi - 1} dt, \text{ with } \Gamma(a) = \int_0^{\infty} t^{a-1} e^{-t} dt$$

Fitting a straight line

As an example of what we discussed, let us consider the case of fitting a straight line, y(x)=a+bx. In that case the chi square is,

$$\chi^2(a,b) = \sum_{i=1}^{N} \left(\frac{y_i - a - bx_i}{\sigma_i} \right)^2$$

We minimize with respect to a and b,

$$0 = \frac{\partial \chi^2}{\partial a} = -2 \sum_{i=1}^{N} \frac{y_i - a - bx_i}{\sigma_i^2}$$

$$0 = \frac{\partial \chi^2}{\partial b} = -2 \sum_{i=1}^{N} \frac{x_i (y_i - a - bx_i)}{\sigma_i^2}$$

It is convenient to define:

$$S \equiv \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \quad S_x \equiv \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} \quad S_y \equiv \sum_{i=1}^{N} \frac{y_i}{\sigma_i^2}$$

$$S_{xx} \equiv \sum_{i=1}^{N} \frac{x_i^2}{\sigma_i^2} \quad S_{xy} \equiv \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2}$$

The equations become,

$$aS + bS_x = S_y$$
$$aS_x + bS_{xx} = S_{xy}$$

And can be readily solved,

$$\Delta \equiv SS_{xx} - (S_x)^2$$
$$a = \frac{S_{xx}S_y - S_xS_{xy}}{\Delta}$$
$$b = \frac{SS_{xy} - S_xS_y}{\Delta}$$

We now want to determine the uncertainties in the values of a,b. To do that we use the formula for the variance of a function,

$$\sigma_f^2 = \sum_{i=1}^{N} \sigma_i^2 \left(\frac{\partial f}{\partial y_i}\right)^2$$

$$\frac{\partial a}{\partial y_i} = \frac{S_{xx} - S_x x_i}{\sigma_i^2 \Delta}$$
$$\frac{\partial b}{\partial y_i} = \frac{Sx_i - S_x}{\sigma_i^2 \Delta}$$

$$\sigma_a^2 = S_{xx}/\Delta$$
$$\sigma_b^2 = S/\Delta$$

Finally, we should compute the incomplete Gamma function to see how likely is the value of chi squared we have.

```
      SUBROUTINE fit(x,y,ndata,sig,mwt,a,b,siga,sigb,chi2,q)
      INTEGER mwt,ndata
      REAL a,b,chi2,q,siga,sigb,sig(ndata),x(ndata),y(ndata)
C     USES gammq
```

Given a set of data points `x(1:ndata)`,`y(1:ndata)` with individual standard deviations `sig(1:ndata)`, fit them to a straight line $y = a + bx$ by minimizing χ^2. Returned are `a`,`b` and their respective probable uncertainties `siga` and `sigb`, the chi-square `chi2`, and the goodness-of-fit probability `q` (that the fit would have χ^2 this large or larger). If `mwt=0` on input, then the standard deviations are assumed to be unavailable: `q` is returned as `1.0` and the normalization of `chi2` is to unit standard deviation on all points.

```
      INTEGER i
      REAL sigdat,ss,st2,sx,sxoss,sy,t,wt,gammq
      sx=0.                                     Initialize sums to zero.
      sy=0.
      st2=0.
      b=0.
      if(mwt.ne.0) then                         Accumulate sums ...
        ss=0.
        do 11 i=1,ndata                         ...with weights
          wt=1./(sig(i)**2)
          ss=ss+wt
          sx=sx+x(i)*wt
          sy=sy+y(i)*wt
        enddo 11
      else
        do 12 i=1,ndata                         ...or without weights.
          sx=sx+x(i)
          sy=sy+y(i)
        enddo 12
        ss=float(ndata)
      endif
      sxoss=sx/ss
      if(mwt.ne.0) then
        do 13 i=1,ndata
          t=(x(i)-sxoss)/sig(i)
          st2=st2+t*t
          b=b+t*y(i)/sig(i)
        enddo 13
```

```
else
    do 14 i=1,ndata
        t=x(i)-sxoss
        st2=st2+t*t
        b=b+t*y(i)
    enddo 14
endif
b=b/st2                                  Solve for a, b, σ_a, and σ_b.
a=(sy-sx*b)/ss
siga=sqrt((1.+sx*sx/(ss*st2))/ss)
sigb=sqrt(1./st2)
chi2=0.                                  Calculate χ^2.

q=1.
if(mwt.eq.0) then
    do 15 i=1,ndata
        chi2=chi2+(y(i)-a-b*x(i))**2
    enddo 15
    sigdat=sqrt(chi2/(ndata-2))          For unweighted data evaluate typical sig us-
    siga=siga*sigdat                         ing chi2, and adjust the standard devia-
    sigb=sigb*sigdat                         tions.
else
    do 16 i=1,ndata
        chi2=chi2+((y(i)-a-b*x(i))/sig(i))**2
    enddo 16
    if(ndata.gt.2) q=gammq(0.5*(ndata-2),0.5*chi2)      Equation (15.2.12).
endif
return
END
```

Summary

- Nonparametric correlation is more robust than linear correlation. Either use Spearman or Kendall's tau.
- Modeling of data, properly done, should provide values for the parameters, errors and statistical significance.