```
ceph
        #grabbing cepheid data and fitting lines
        define NGC_number ?
        define filen n$NGC_number.dat
        data $filen
#       data n925.dat

        read { P 1 I 2 }
        set logP = lg(P)

#       device x11
        device postencap '$!NGC_number.ps'


        ctype black
        era
        lim 0.5 2.5 28 20
        xlabel log(Period)
        ylabel <I>
        toplabel NGC $NGC_number with fit line
        box
        points logP I
        #Using from Freedman, W. L., et al. 2001, ApJ, 553, 47
        #M<I> = -2.962 log( P ) - 1.942 (sigma = 0.11)

        #minimising chi square:
        #looping over offsets
        define best 10000000
        define bestnum 0
        do ofst = 25, 35, 0.01 {
                set chivect = I - ($ofst - 2.962*logP - 1.942)
                set chivs = chivect**2
                define tempsum sum(chivs)
                if ($tempsum < $best) {
                        define best $tempsum
                        define bestnum $ofst
                }
        }

        echo distance modulus: $bestnum

        #calculate error on this fit
        set chivect = I - ($bestnum - 2.962*logP - 1.942)
        stats chivect chivectavg chivectsig crap
        echo error in fit: $chivectsig

        #plot this line onto graph
        set plotlogP = 0, 2.5, 0.0001
        set plotI = $bestnum - 2.962*plotlogP - 1.942
```

```
        connect plotlogP plotI

        lim 0 10 0 10
        relocate 5 2
        label m-M = $bestnum +/-
        relocate 6 1.5
        label $chivectsig



inc
        set NGC = { 224 598 925 1365 1425 2090 2403 2541 3031 3198 3319 3351 3368 3621 3627 4414 4535 4536 4548 4725 7331
}
        data incs
        read { ab 5 }
        set ba = ab**-1
        define abuncert 0.25
        set abplus = ab + $abuncert
        set abminus = ab - $abuncert
        set baplus = abplus**-1
        set baminus = abminus**-1


        set inc = acosd(ba)
        set incunc = (abs(acosd(baplus)-2*acosd(ba))+abs(acosd(baminus)-acosd(ba)))/2



w
        #doing velocity conversion to normal units and inclination correction
        #   AND error propagation

        #read in w data
        data 2.dat
        read { logw 5 logwunc 6 }

        set w = 10**logw
        set whigh = 10**(logw+logwunc)
        set wlow = 10**(logw-logwunc)
        set wunc = (abs(whigh-w)+abs(wlow-w))/2

        set v = w / sind(inc)
        set vunc = sqrt(((wunc/sind(inc))**2)+((-incunc*cosd(inc)/((sind(inc))**2))**2))



fitting
        #using lsq2, do the fit!

        data 2final.dat
```

```
        read { num 1 M 2 Munc 3 v 4 vunc 5 }
        set logv = lg(v)
        set logvhigh = lg(v+vunc)
        set logvlow = lg(v-vunc)
        set logvunc = (abs(logvhigh-logv)+abs(logv-logvlow))/2

#       device x11
        device postencap 'TF.ps'
        era
        ctype black
        lim 2.2 2.8 -19.5 -24
        box
        xlabel log(v)
        ylabel M_I
        points logv M
        ERRORBAR logv M logvunc 1
        ERRORBAR logv M logvunc 3
        ERRORBAR logv M Munc 2
        ERRORBAR logv M Munc 4

        #x is logv y is M
        #sigma_x = logvunc, sigma_y Munc

        set logvfit = 2.2, 2.8, .001
        lsq logv M logvfit Mfit rms
        connect logvfit Mfit

        toplabel M_I = (-7.23 +/- 0.83)log(V) - (3.24 +/- 2.15)

        #lim 0 10 0 10
        #relocate -1.3 10.3
        #label M_I=lgV($a+/-$sig_a)+($b+/-$sig_b)


lsq3 6
        #re-doing their lsq2, making.... lsq3!!!

        # ... just kidding



oldstuff
        #outdated
        #and bad

        #inclination part
        data incs
        read { maj 3 min 4 }
        set inc = acosd(min/maj)
```

```
#estimate 10% error on inclination?
set incunc = inc*.1

#calculate correction factor 1/sind(inc)
set corr = 1/sind(inc)

#calculate uncert on correction factor
set corrunc = (1/sind(inc+incunc) + 1/sind(inc-incunc) - 2*corr)/2

#calculate log(correction factor) and log(corrunc)
set logcorr = lg(corr)
set logcorrunc = abs(lg(corr+corrunc) + lg(corr-corrunc) - 2*lg(corr))/2
#propagate error from log(w_20)
data 2.dat
read { logW 5 logWunc 6 }
set W = 10**logW
set Wunc = abs(10**(logW+logWunc) + 10**(logW-logWunc) - 2*10**(logW))/2

set Wtotunc = sqrt((Wunc*corr)**2 + (corrunc*W)**2)
set Wtot = W*corr

set logWtot = lg(Wtot)
set logWtotunc = (abs(lg(Wtot-Wtotunc) - lg(Wtot)) + abs(lg(Wtot+Wtotunc) - lg(Wtot)))/2
```