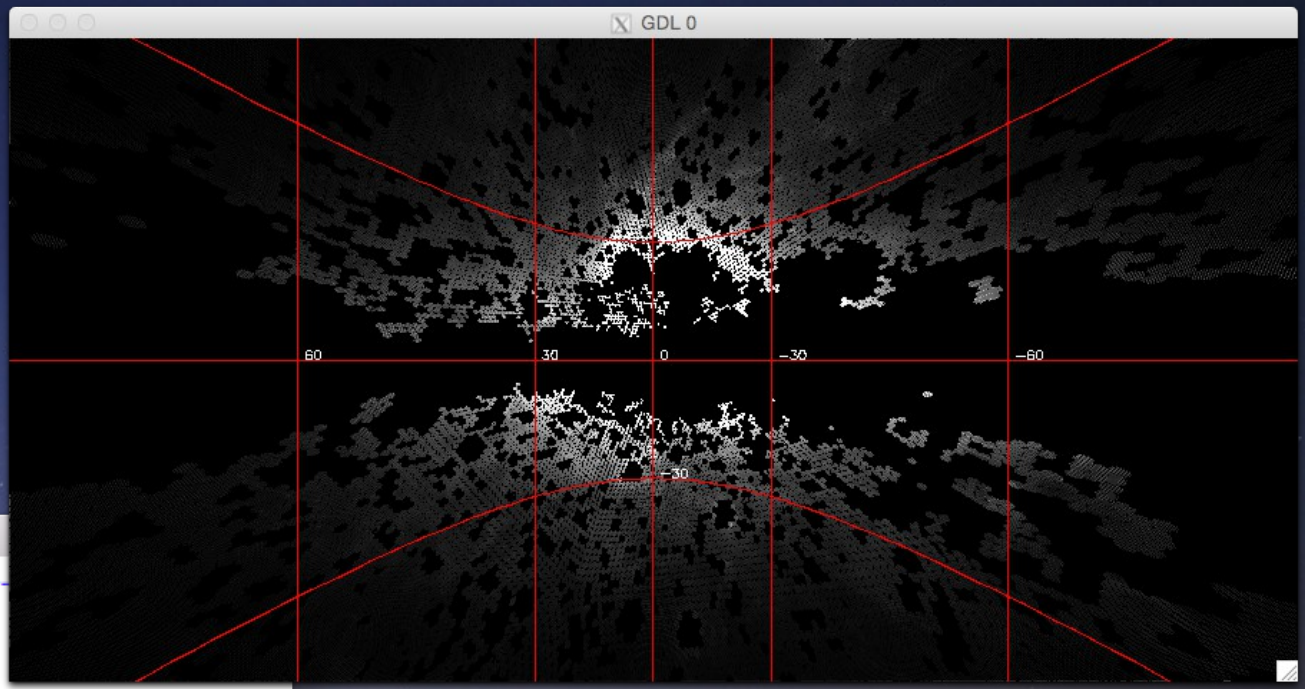
# Image Analysis

Jayant Murthy
The Indian Institute of Astrophysics
www.iiap.res.in
jmurthy@yahoo.com

# Converting between projections




2. more

```
im=mrdfits("data_files/hlsp_uv-bkgd_galex_map_allsky_fuv_v1_
r)
nx = sxpar(hdr, "NAXIS1")
ny = sxpar(hdr, "NAXIS2")
x = lindgen(nx, ny) mod nx
y = lindgen(nx, ny)/nx
extast, hdr, astr
xy2ad,x,y,astr,gl,gb
sxaddpar,hdr,"CTYPE1","GLON-TAN"
sxaddpar,hdr,"CTYPE2","GLAT-TAN"
extast,hdr,astr2
ad2xy,gl,gb,astr2,x2,y2
x2(where(finite(x2) eq 0)) = 0
y2(where(finite(y2) eq 0)) = 0
x2 = 0 > x2 < (nx - 1)
y2 = 0 > y2 < (ny - 1)
im2=fltarr(nx,ny)
for i=0,nx-1 do for j=0,ny-1 do im2(x2(i,j)>0,y2(i,j)>0)=im(x(i,j),y(i,j))
window,xs=900,ys=450
tv,bytscl(rebin(im2,900,450),0,2000)

plot,/nodata,/noerase,pos=[0,0,1,1],xstyle=5,ystyle=5,[0,3600],[0,1800]
oplot,[1800,1800],[0,1800],col=255
```

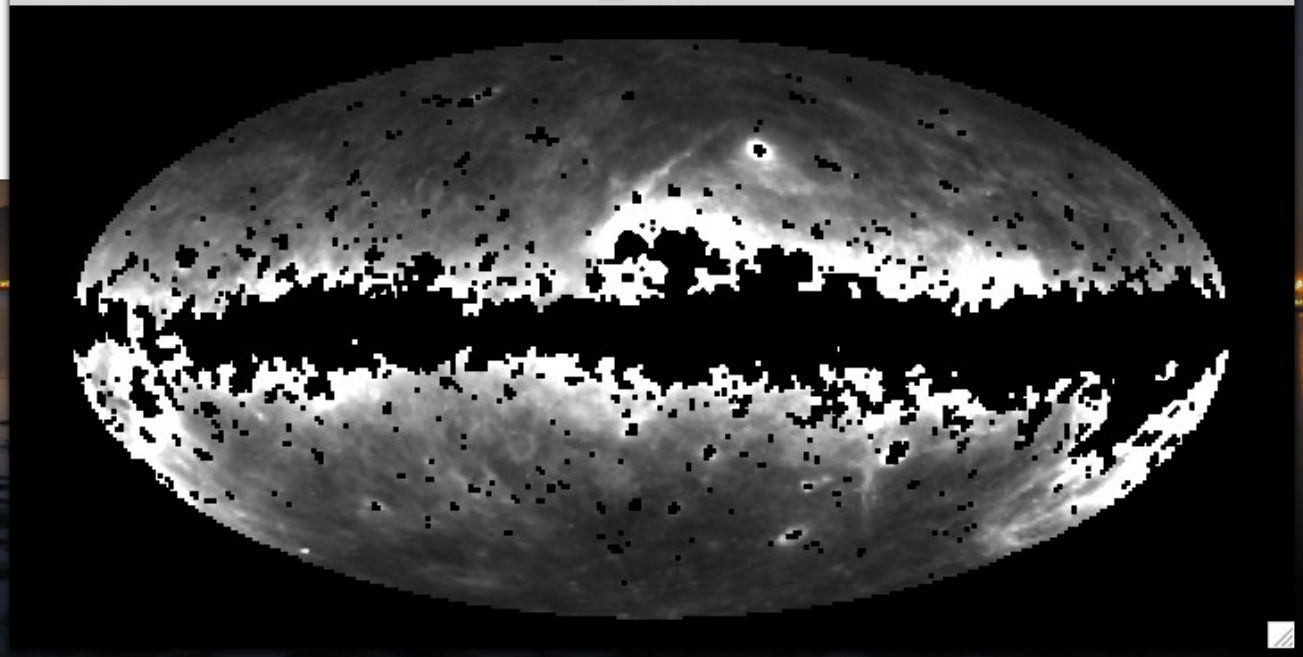
cnvt_ait_to_tan.pro

# Rebinning

2. bash

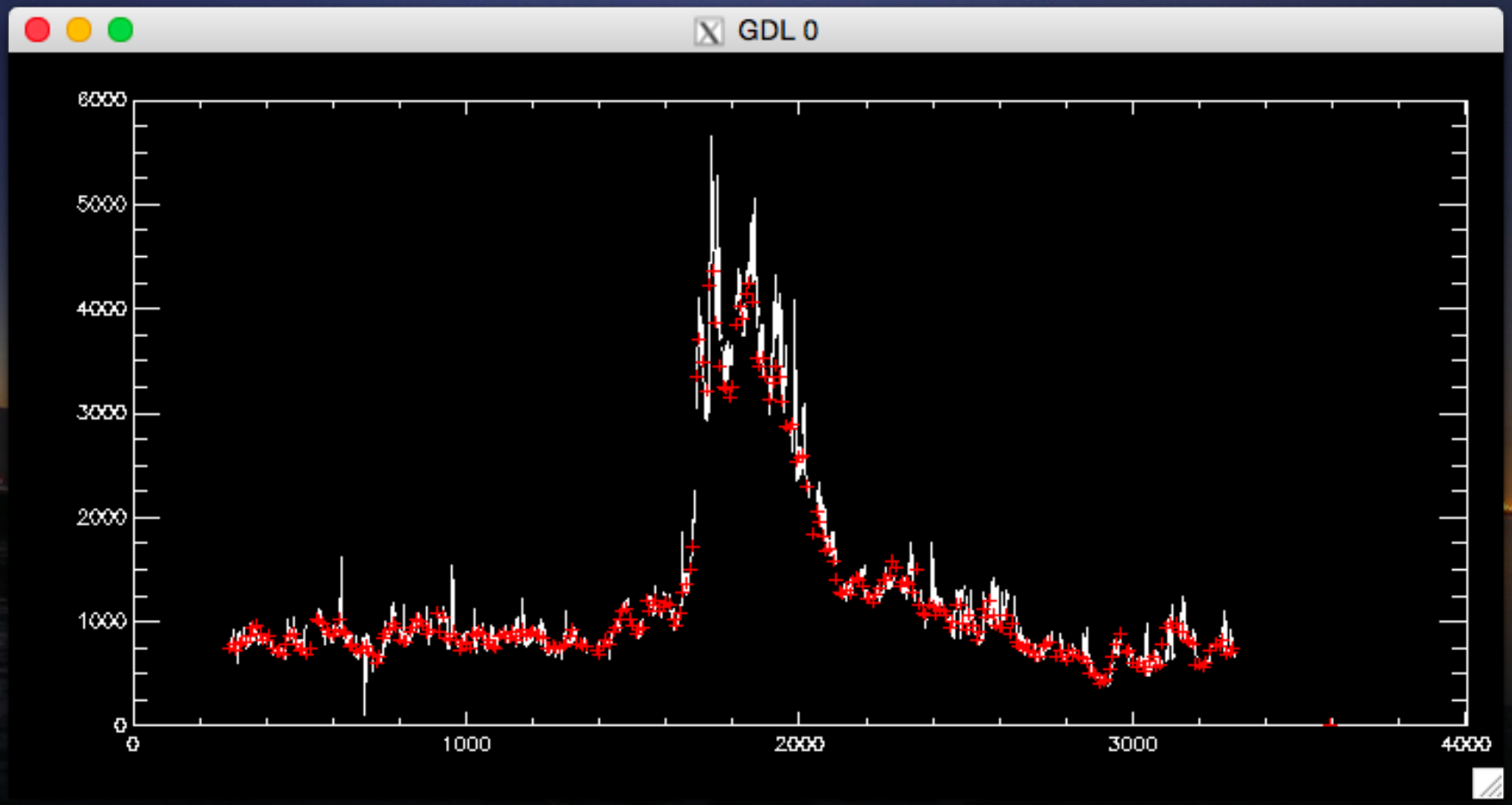
```
 im=mrdfits("data_files/hlsp_uv-bkgd_galex_map_allsky_fuv_v1_skymap.fits.gz",0,h
dr)

nx = 3600
ny = 1800
nx2 = 360
ny2 = 180
new_im = fltarr(nx2, ny2)
for i = 0, nx2 - 2 do begin
  for j = 0, ny2 - 2 do begin
   mx1 = i*10
   mx2 = (i + 1)*10 - 1
   my1 = j*10
   my2 = (j + 1)*10 - 1
   arr = im(mx1:mx2, my1:my2)
   q = where((finite(arr) eq 1) and (arr gt 0), nq)
   if (nq gt 0) then new_im(i, j) = mean(arr(q)) else new_im(i, j) = -9999
 endfor
endfor

window,xs=720,ys=360
rim=rebin(new_im,720,360)
tv,bytscl(rim,0,max(rim)/4)
end
$
```

GDL 0



# Data Cuts

- GDL> plot,im(*,1200)
- GDL> oplot,findgen(360)*10,new_im(*,120),col=255,/psym



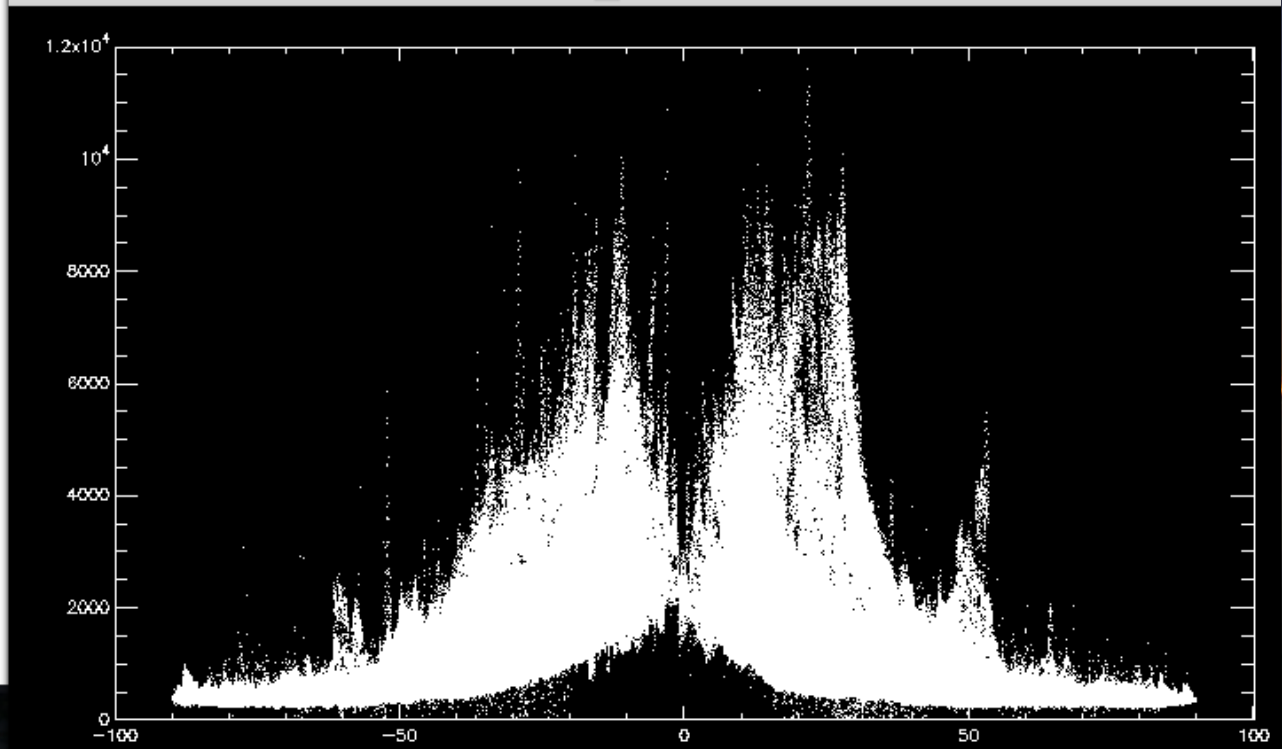
# Analysis

- UV light from Galactic dust so should be related to the latitude.

2. vim

```
im=mrdfits("data_files/hlsp_uv-bkgd_galex_map_allsky_fuv_v1_skymap.fits.gz",0,hd
r)
nx = sxpar(hdr, "NAXIS1")
ny = sxpar(hdr, "NAXIS2")
x = lindgen(nx, ny) mod nx
y = lindgen(nx, ny)/nx
extast, hdr, astr
xy2ad,x,y,astr,gl,gb

plot,gb,im,psym=3
end
~
~
~
~
~
~
~
~
~
~
~
~
"cosecan_lat.pro" 10L, 247C written
```

GDL 0

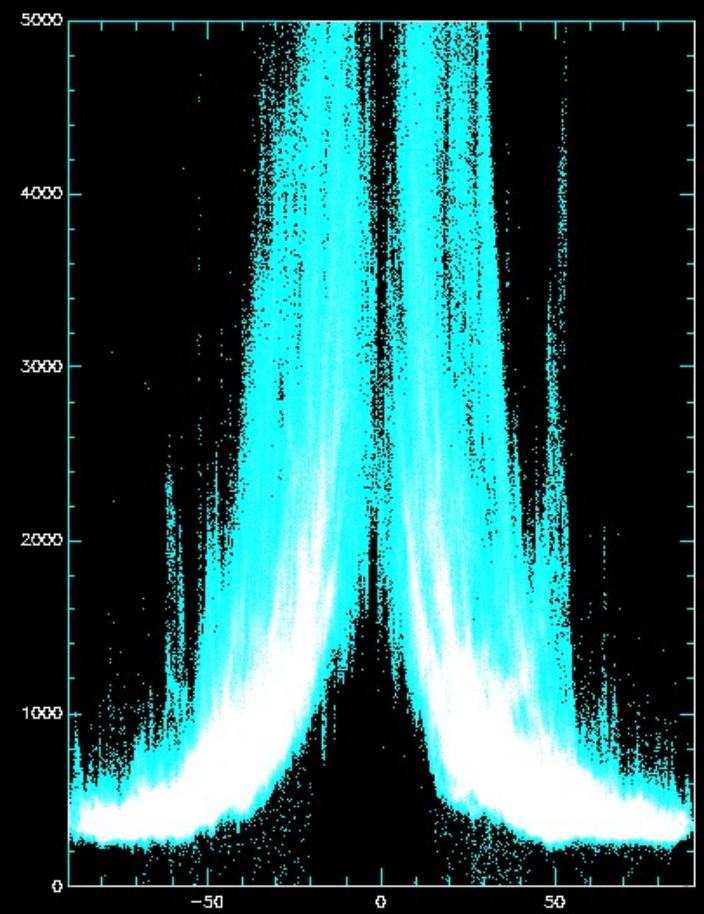


# Density Plot

2. vim

```
im=mrdfits("data_files/hlsp_uv-bkgd_galex_map_allsky_fuv_v1_skymap.fits.gz",0,hd
r)
nx = sxpar(hdr, "NAXIS1")
ny = sxpar(hdr, "NAXIS2")
x = lindgen(nx, ny) mod nx
y = lindgen(nx, ny)/nx
extast, hdr, astr
xy2ad,x,y,astr,gl,gb

plot,gb,im,psym=3

;Density plot
grid=fltarr(360,500)
for i=0,nx-1 do for j=0,ny-1 do begin
   xaxis = fix((90 + gb(i,j))*2)
   yaxis = 0 > fix(im(i,j)/10) < 499
   grid(xaxis, yaxis) = grid(xaxis, yaxis) + 1
endfor

plot,[-90,90],[0,5000],pos=[100,100,460,600],/dev,/nodata,xrange=[-90,90],$
                       yrange=[0,5000],xstyle=1,ystyle=1
oplot,gb,im,psym=3
tv,bytscl(grid,0,50),100,100,channel=1
```

GDL 0



# Cosecant plots

- Anything related to our Galaxy will be fit by a cosecant law.
- We can solve using least squares.

- y = a*csc(abs(gl)) + b
- x = 1/sin(abs(gl))
- y = ax + b

# Least Squares Function

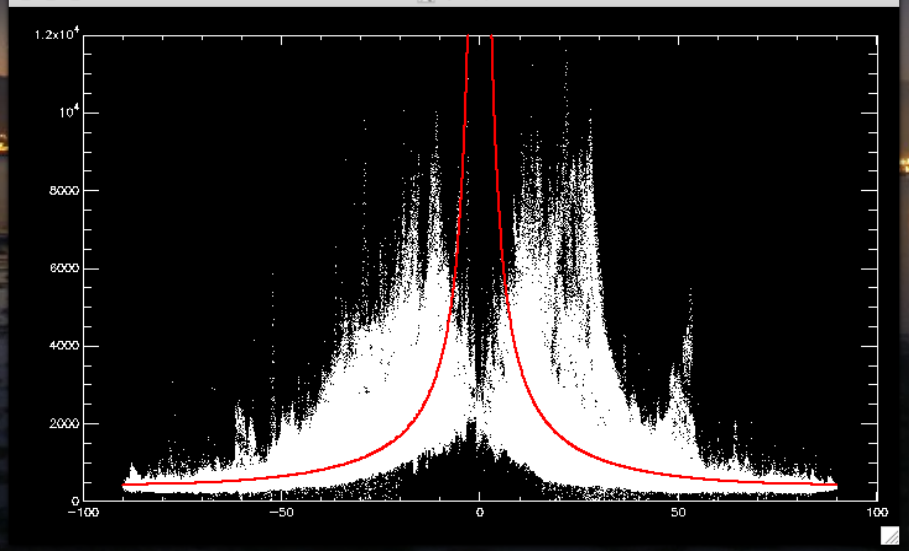
2. bash

```
$cat least_squares.pro
function least_squares,x, y, sigma

nelems = n_elements(x)
if (nelems eq 0) then begin
 print,"No input variable defined"
 return,[0,0]
endif

if (n_elements(y) ne nelems)then begin
 print,"Number of elements in dependent variable must match"
 return,[0,0]
endif

if (n_elements(sigma) eq 0) then sigma = fltarr(nelems) + 1

;Begin calculation
yx = total(x*y/sigma^2)
sumyx = total(x/sigma^2)*total(y/sigma^2)
xsqr = total(x*x/sigma^2)
sum_x_sqr = total(x/sigma^2)*total(x/sigma^2)
tot_sig = total(1/sigma^2)
a = (yx*tot_sig - sumyx)/(xsqr*tot_sig - sum_x_sqr)

b = (total(y/sigma^2) - total(x/sigma^2)*a)/tot_sig
return,[a,b]
end
$
```

2. bash

```
$fg
vi least_squares.pro
$cat least_squares.com
im=mrdfits("data_files/hlsp_uv-bkgd_galex_map_allsky_fuv_v1_skymap.fits.gz",0,hd
r)
nx = sxpar(hdr, "NAXIS1")
ny = sxpar(hdr, "NAXIS2")
x = lindgen(nx, ny) mod nx
y = lindgen(nx, ny)/nx
extast, hdr, astr
xy2ad,x,y,astr,gl,gb
csc = 1./sin(abs(gb)/!radeg)
q = where((finite(im) eq 1) and (im gt 0) and (finite(csc) eq 1) and (abs(gb) gt
 10))
.run programs/least_squares
par = least_squares(csc(q),im(q))
plot,gb,im,psym=3

cb = findgen(1800)*.1-90
xp = 1./abs(sin(cb/!radeg))
model = par[0]*xp + par[1]
oplot,cb,model,thick=2,col=255
print,par
stop
$
```

GDL 0

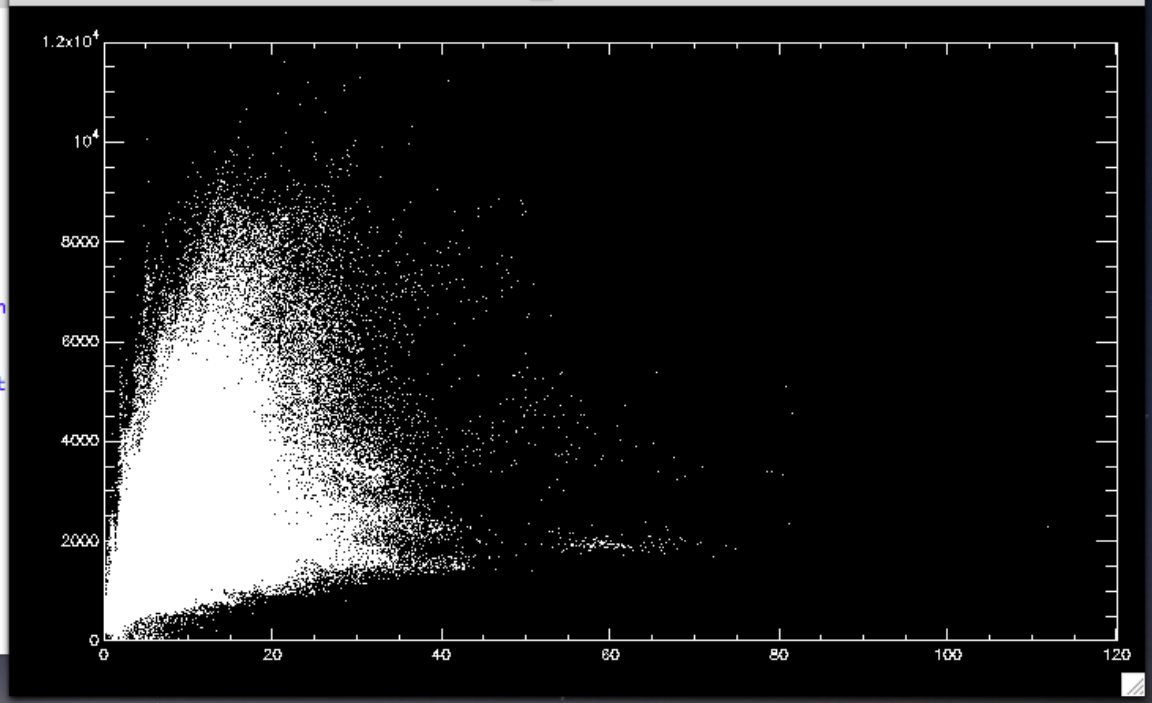


# UV/IR Correlations

1. bash

```
*** PLPLOT WARNING ***
You said you want pthreads, but they are not available.
% Compiled module: CORRELATE.
% Compiled module: MEAN.
          nan
     0.765580
GDL>
$more programs/ir_uv_correlation.com
; GDL Version 0.9.5 (darwin x86_64 m64)
; Working directory: /Users/jayanth/user/education/course/data_analysis
fuv=mrdfits("data_files/hlsp_uv-bkgd_galex_map_allsky_fuv_v1_skymap.fits.gz",0,h
dr)
;MRDFITS: Image array (3600,1800)  Type=Real*4
ir=mrdfits("data_files/hlsp_uv-bkgd_cobe-iras_map_allsky_100micron_v1_skymap.fit
s",0,ir_hdr)
;MRDFITS: Image array (3600,1800)  Type=Real*4
plot,ir,fuv,psym=3
print,correlate(ir,fuv)
;          nan
q=where((ir gt 0) and (fuv gt 0) and (finite(fuv) gt 0) and (finite(ir) gt 0))
plot,ir(q),fuv(q),psym=3
print,correlate(ir(q),fuv(q))
;     0.765580
$
```

GDL 0



# Density function

2. bash

```
print,string(13b), "ETC: ",(systime(1) - start_time)/i*(nx - 1),$
        format = '(a1, a5, i10, $)'
  for j=0,ny-2 do begin
    y1 = ymin + j*yinc
    y2 = ymin + (j + 1)*yinc
    q = where((ys(i1:i2) ge y1) and (ys(i1:i2) lt y2), nq)
    grid(i, j) = grid(i, j) + nq
  endfor
endfor

return, grid
end
$more density_slow.pro.pro
density_slow.pro.pro: No such file or directory
$more density_slow.pro
function density, xvar, yvar, xinc, yinc, xmin = xmin, xmax = xmax, $
                ymin = ymin, ymax = ymax

if (n_elements(xmin) eq 0) then xmin = min(xvar)
if (n_elements(xmax) eq 0) then xmax = max(xvar)
if (n_elements(ymin) eq 0) then ymin = min(yvar)
if (n_elements(ymax) eq 0) then ymax = max(yvar)

nx = fix(xmax/xinc) + 1
ny = fix(ymax/yinc) + 1

;Density plot
grid=fltarr(nx, ny)
start_time = systime(1)
for i=0,nx-2 do begin
  print,string(13b), "ETC: ",(systime(1) - start_time)/i*(nx - 1),$
        format = '(a1, a5, i10, $)'
  for j=0,ny-2 do begin
   x1 = xmin + i*xinc
   x2 = xmin + (i + 1)*xinc
   y1 = ymin + j*yinc
   y2 = ymin + (j + 1)*yinc
   q = where((xvar ge x1) and (xvar lt x2) and $
                (yvar ge y1) and (yvar lt y2), nq)
   grid(i, j) = grid(i, j) + nq
  endfor
endfor

return, grid
end
$
```

2. bash

```
;MRDFITS: Image array (3600,1800)  Type=Real*4
plot,ir,fuv,psym=3
print,correlate(ir,fuv)
;          nan
q=where((ir gt 0) and (fuv gt 0) and (finite(fuv) gt 0) and (finite(ir) gt 0))
plot,ir(q),fuv(q),psym=3
print,correlate(ir(q),fuv(q))
;     0.765580
.run programs/density
grid=density(ir(q),fuv(q),0.1,10,xmin=0,xmax=12,ymin=0,ymax=5000)
$more density.pro
function density, xvar, yvar, xinc, yinc, xmin = xmin, xmax = xmax, $
                ymin = ymin, ymax = ymax

if (n_elements(xmin) eq 0) then xmin = min(xvar)
if (n_elements(xmax) eq 0) then xmax = max(xvar)
if (n_elements(ymin) eq 0) then ymin = min(yvar)
if (n_elements(ymax) eq 0) then ymax = max(yvar)

nx = fix(xmax/xinc) + 1
ny = fix(ymax/yinc) + 1

;Density plot
grid=fltarr(nx, ny)
s = sort(xvar)
xs = xvar(s)
ys = yvar(s)
start_time = systime(1)
for i=0,nx-2 do begin
  x1 = xmin + i*xinc
  x2 = xmin + (i + 1)*xinc
  i1 = min(where(xs ge x1))
  i2 = min(where(xs ge x2))
print,string(13b), "ETC: ",(systime(1) - start_time)/i*(nx - 1),$
        format = '(a1, a5, i10, $)'
  for j=0,ny-2 do begin
    y1 = ymin + j*yinc
    y2 = ymin + (j + 1)*yinc
    q = where((ys(i1:i2) ge y1) and (ys(i1:i2) lt y2), nq)
    grid(i, j) = grid(i, j) + nq
  endfor
endfor

return, grid
end
$
```

# Multicolour