Data reduction II
Photometry with IRAF

Ingrid Pelisoli

Research workshop on evolved stars
09.09.2019

Why data *reduction*?

- We need to subtract – or reduce – instrumental effects and background contamination.

Reducing instrumental effects:

- **BIAS**: image with zero exposure time.
  Estimate of the real zero of the CCD.

- **FLAT**: image of a uniformly illuminated surface.
  Estimate sensibility difference throughout the CCD.

- **DARK**: image with the same exposure time of the science image with the shutter closed.
  Estimate the level of background current.

More is better

- Each of the counts on the images has an associated uncertainty.
- If we take n images, each with an uncertainty σ_i, the uncertainty on the average will be $\sigma_i/\sqrt{n}$.
- Therefore, the first step in data reduction is to calculate the average for BIAS, FLAT, and DARK images.

More is better

- Each of the counts on the images has an associated uncertainty.
- If we take n images, each with an uncertainty σ_i, the uncertainty on the average will be $\sigma_i/\sqrt{n}$.
- Therefore, the first step in data reduction is to calculate the average for BIAS, FLAT, and DARK images.

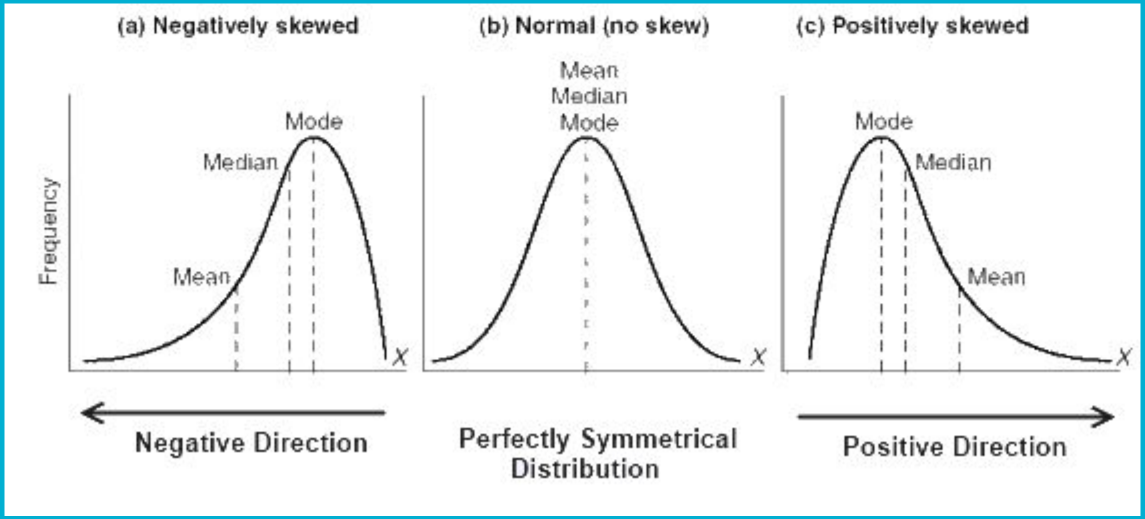
  BIAS: not available.

  FLAT: master flat has already been calculated.

  DARK: we need to calculate the median flat.

Mean or median?

- Either of those can be representative of a distribution – which one should we use?
- The mean is sensitive to outliers – the median is robust against outliers.
- The mean is not descriptive for skewed distributions.
- Give preference to the median!



IRAF – Image Reduction and Analysis Facility

```
pelisoli@octans:...2/envs/iraf27/iraf

     This is the EXPORT version of IRAF V2.16 supporting PC systems.


  Welcome to IRAF.  To list the available commands, type ? or ??.  To get
  detailed information about a command, type 'help <command>'.  To run  a
  command  or  load  a  package,  type  its name.   Type  'bye' to exit a
  package, or 'logout' to get out  of the CL.    Type 'news' to find  out
  what is new in the version of the system you are using.

  Visit http://iraf.net if you have questions or to report problems.

  The following commands or packages are currently defined:

  (Updated on 2013-12-13)

      adccdrom.   deitab.     images.     mtools.     softools.   upsqiid.
      cfh12k.     esowfi.     kepler.     nfextern.   sqiid.      utilities.
      cirred.     finder.     language.   noao.       stecf.      vo.
      ctio.       fitsutil.   lists.      obsolete.   stsdas.     xdimsum.
      cutoutpkg.  gemini.     mem0.       plot.       system.     xray.
      dataio.     gmisc.      mscdb.      proto.      tables.
      dbms.       guiapps.    mscred.     rvsao.      ucsclris.

ecl>
```

Using IRAF for the first time

- To start IRAF: open an xgterm terminal and type `cl`.
- Likely this will issue a warning: *no login.cl found in login directory*.
- The file login.cl contains the default configuration for IRAF; you should create it before using it for the first time.
- Exit IRAF by typing `logout`, and then create the login.cl by typing `mkiraf`; choose terminal type xgterm.
- Edit the file login.cl according to your preferences, mainly:

  set editor = emacs
- Now start iraf again.

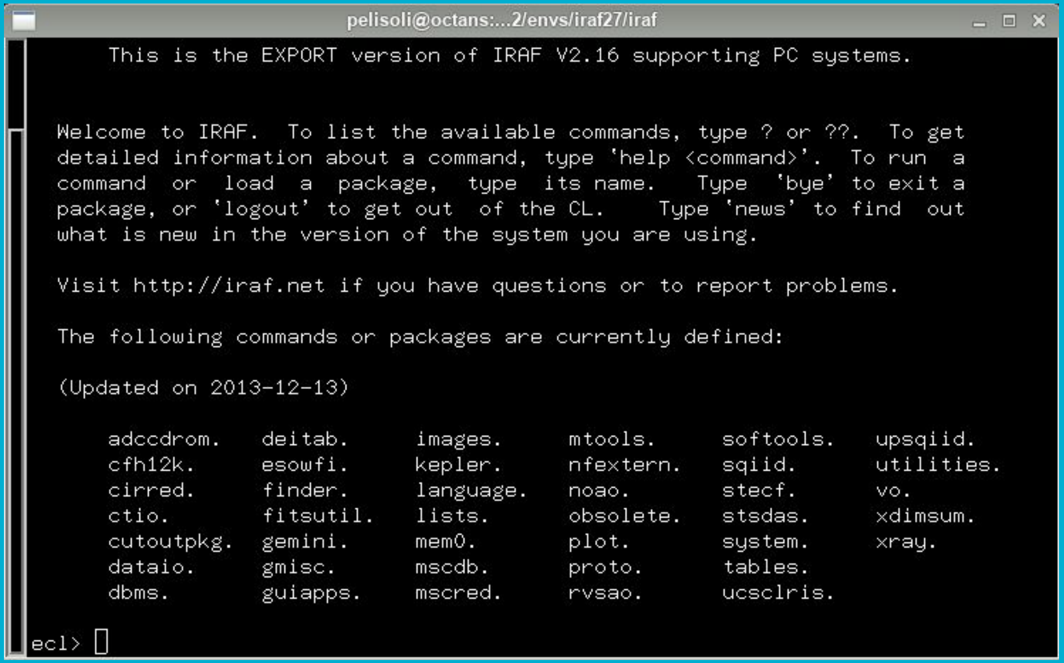
IRAF – Image Reduction and Analysis Facility

Some basic commands:

`epar [task]` → edit task parameters.

`:wq` → write the parameters and exit.

`:go` → execute the task.



IRAF – Image Reduction and Analysis Facility

- A really useful tool is the task imexamine, which allows to analyse fits images.
- It can be used as a quick-look tool during observing runs.
- To use it, you will need to display the images in ds9. To open it, type:

```
!ds9 &
```

IRAF – Image Reduction and Analysis Facility

- A really useful tool is the task imexamine, which allows to analyse fits images.
- It can be used as a quick-look tool during observing runs.
- To use it, you will need to display the images in ds9. To open it, type:

```
!ds9 &
```

`!` — Tells IRAF this is an external command.

`&` — Sends it to background so you can still use command lines.

IRAF – Image Reduction and Analysis Facility

- A really useful tool is the task imexamine, which allows to analyse fits images.
- It can be used as a quick-look tool during observing runs.
- To use it, you will need to display the images in ds9. To open it, type:

```
!ds9 &
```

- To display an image, do `display [image name]`
- Run imexamine (just type `imexam` on terminal).
- You will notice the cursor turns into a circle in ds9 – this means imexamine is activated.
- Some basic imexamine commands:

`l` – plot the counts on the selected line
`c` - plot the counts on the selected column
`r` - display radial profile
a - show counts, sky, FWHM, etc. on screen
e - display contours

IRAF – Image Reduction and Analysis Facility

- We'll use the package

  `noao.imred.ccdred`

  for the data reduction, and

  `noao.digiphot.daophot`

  for the photometry.

- Load each part of the packages by typing their name followed by enter.

pelisoli@octans:...2/envs/iraf27/iraf

```
ecl> noao
      artdata.      digiphot.     nobsolete.    onedspec.
      astcat.       focas.        nproto.       rv.
      astrometry.   imred.        observatory   surfphot.
      astutil.      mtlocal.      obsutil.      twodspec.

noao> imred
      argus.      crutil.     echelle.    iids.       kpnocoude.  specred.
      bias.       ctioslit.   generic.    irred.      kpnoslit.   vtel.
      ccdred.     dtoi.       hydra.      irs.        quadred.

imred> ccdred
      badpiximage       ccdmask           flatcombine       mkskyflat
      ccdgroups         ccdproc           mkfringecor       setinstrument
      ccdhedit          ccdtest.          mkillumcor        zerocombine
      ccdinstrument     combine           mkillumflat
      ccdlist           darkcombine       mkskycor

ccdred> 
```

Master flat

- The master flats have already been created, but it is good practice to inspect them.
- Open ds9:

```
!ds9 &
```

- Display the flat:

```
display masterflat-R.fit
```

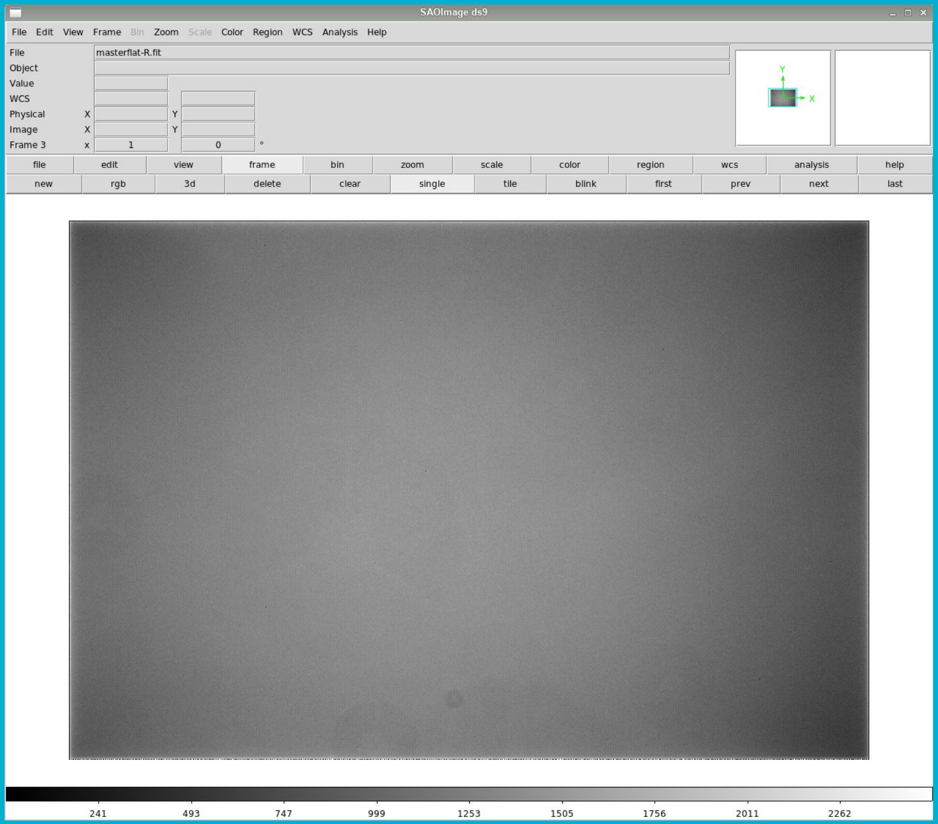
- Plot the flat:

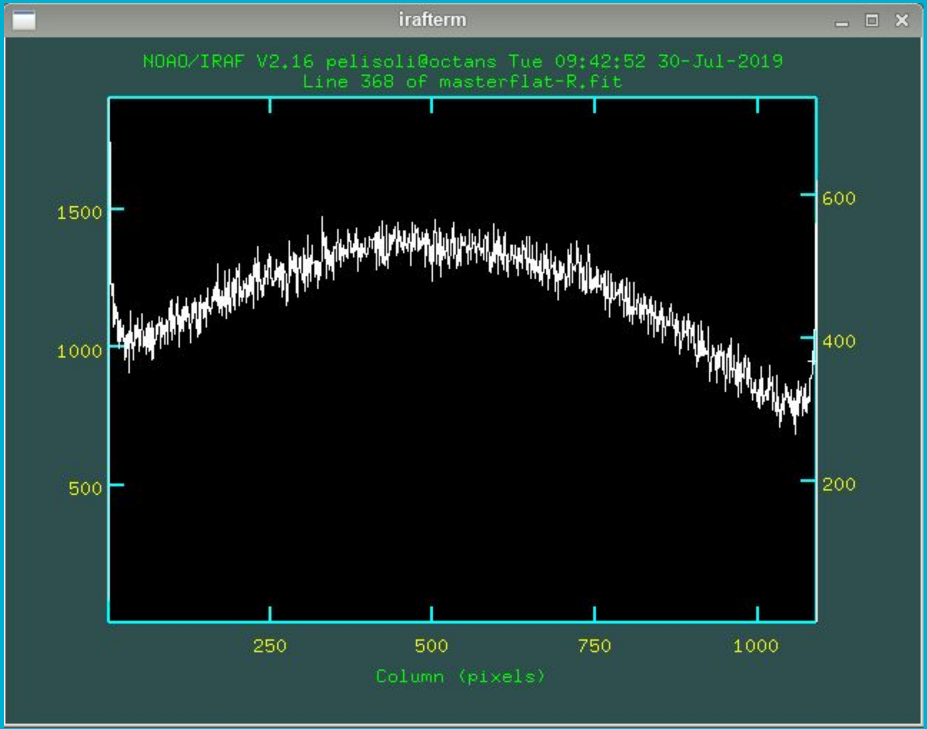
```
implot masterflat-R.fit
```

- Check image statistics:

```
imstat masterflat-R.fit
```

Master flat


SAOImage ds9
File Edit View Frame Bin Zoom Scale Color Region WCS Analysis Help
File masterflat-R.fit
Object
Value
WCS
Physical
Image
Frame 3
file edit view frame bin zoom scale color region wcs analysis help
new rgb 3d delete clear single tile blink first prev next last
241 493 747 999 1253 1505 1756 2011 2262


irafterm
NOAO/IRAF V2.16 pelisoli@octans Tue 09:42:52 30-Jul-2019
Line 368 of masterflat-R.fit
1500
1000
500
600
400
200
250 500 750 1000
Column (pixels)

Creating a master dark

- What is the exposure time of the images we will analyse?
  Check the header!
  `imhead [image name] lo+`

- Which dark images should we use?
  `imhead df-* lo+ | grep EXPTIME`
- Create a list (text document) containing the names of the dark frames using the same exposure time as the science images.

Creating a master dark

@[name of the list you've created]

```
pelisoli@octans:...2/envs/iraf27/iraf

                              I R A F
                Image Reduction and Analysis Facility
PACKAGE = ccdred
   TASK = darkcombine

input   =                  @dark  List of dark images to combine
(output =                  Dark)  Output dark image root name
(combine=                median)  Type of combine operation
(reject =               sigclip)  Type of rejection
(ccdtype=                      )  CCD image type to combine
(process=                    no)  Process images before combining?
(delete =                    no)  Delete input images after combining?
(clobber=                    no)  Clobber existing output image?
(scale  =              exposure)  Image scaling
(statsec=                      )  Image section for computing statistics
(nlow   =                     0)  minmax: Number of low pixels to reject
(nhigh  =                     1)  minmax: Number of high pixels to reject
(nkeep  =                     1)  Minimum to keep (pos) or maximum to reject (neg)
(mclip  =                   yes)  Use median in sigma clipping algorithms?
(lsigma =                    5.)  Lower sigma clipping factor
(hsigma =                    5.)  Upper sigma clipping factor
(rdnoise=                    0.)  ccdclip: CCD readout noise (electrons)
(gain   =                   1.3)  ccdclip: CCD gain (electrons/DN)
(snoise =                    0.)  ccdclip: Sensitivity noise (fraction)
(pclip  =                  -0.5)  pclip: Percentile clipping parameter
(blank  =                    0.)  Value if there are no pixels
(mode   =                    ql)
```

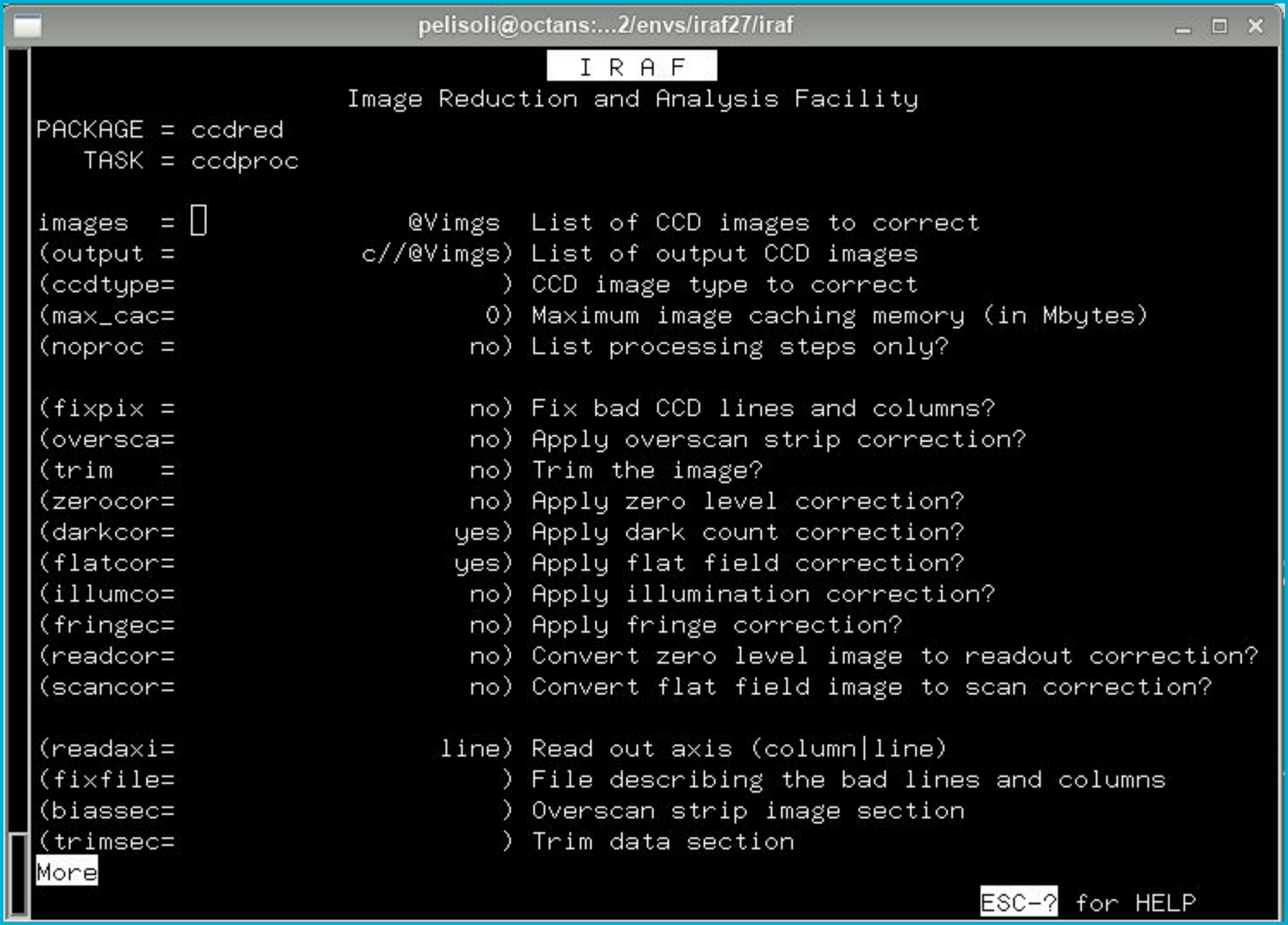
Reducing the science images

- We have images on two different filters: R or V.
  You have to use the correct master flat for each of them.
- Make a list containing the R images, and another containing the V images, e.g.

```
ls Cyg2*R*.fit > Rimgs
ls Cyg2*V*.fit > Vimgs
```

- Use the task `ccdproc` to divide the images by the flat and subtract the dark current. Do it separately for R and V images.

Reducing the science images



Reducing the science images

Photometry

- Now that the images have been reduced, we can perform photometry.
- The first step is to run the task `daofind`, which will find the stars in our image.
- There are a few parameters we need to measure in our image to best setup daofind: the sky and the **F**(ull)**W**(idth)**H**(alf)**M**(aximum)
- For that, display an image at the beginning of the exposure, middle, and end:

```
display cCyg2R001.fit 1
display cCyg2R111.fit 2
display cCyg2R223.fit 3
```

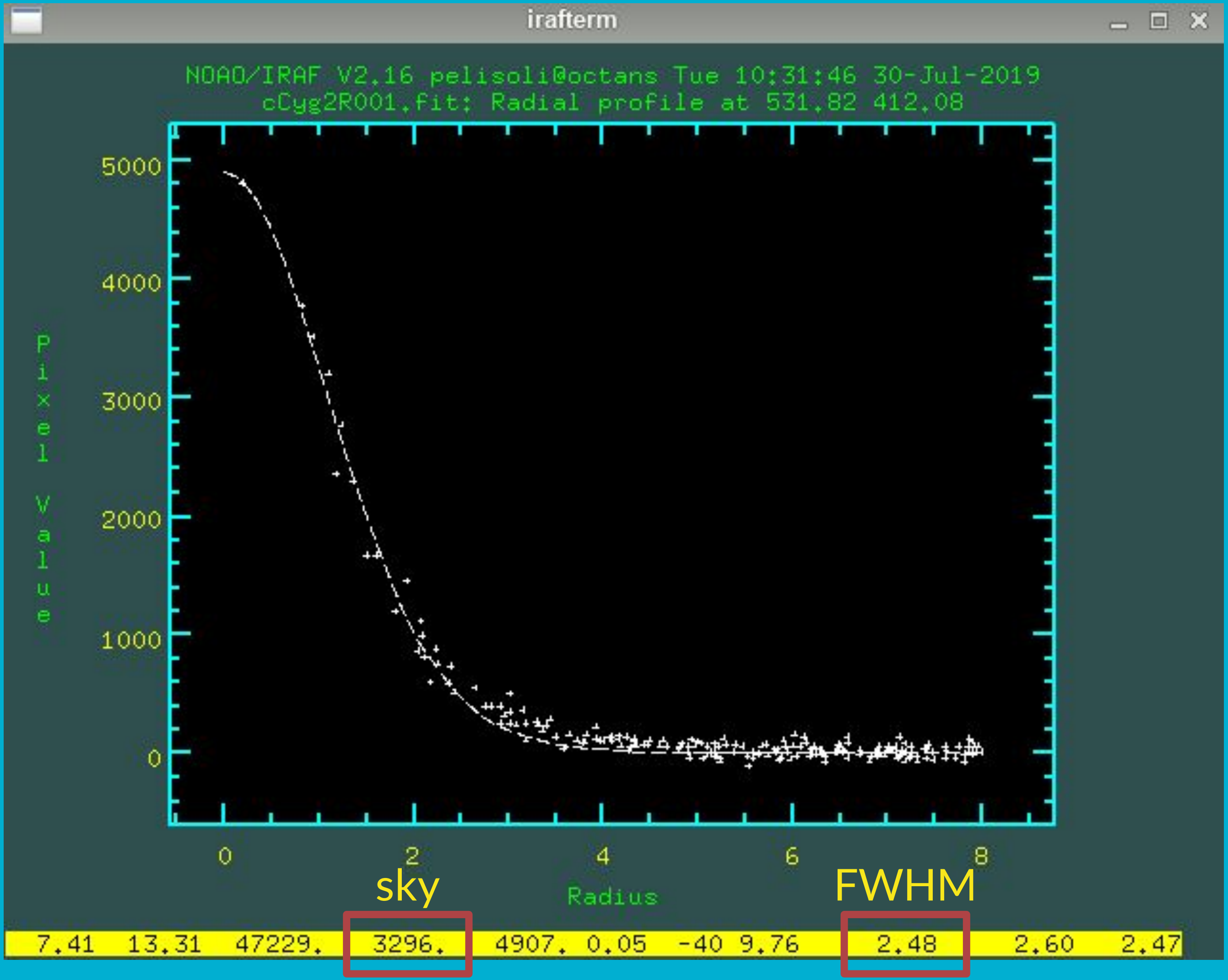
- Use the task `imexamine` – choose a relatively bright near the centre of the image. Centre the cursor on this star.

r → display the radial profile

e → show contours

a → write measurements to the screen

Photometry



irafterm
NOAO/IRAF V2.16 pelisoli@octans Tue 10:31:46 30-Jul-2019
cCyg2R001.fit: Radial profile at 531.82 412.08
5000
4000
3000
2000
1000
0
Pixel Value
0
2
4
6
8
Radius
sky
FWHM
7.41 13.31 47229. 3296. 4907. 0.05 -40 9.76 2.48 2.60 2.47

Photometry

- Check the sky values in the three images. We will use this to set our initial guess for the background. The value of sigma is in turn the square-root of the background (assuming Poissonic noise).

  If the values are very different, use the median; if they are similar, use the mean.

  E. g.

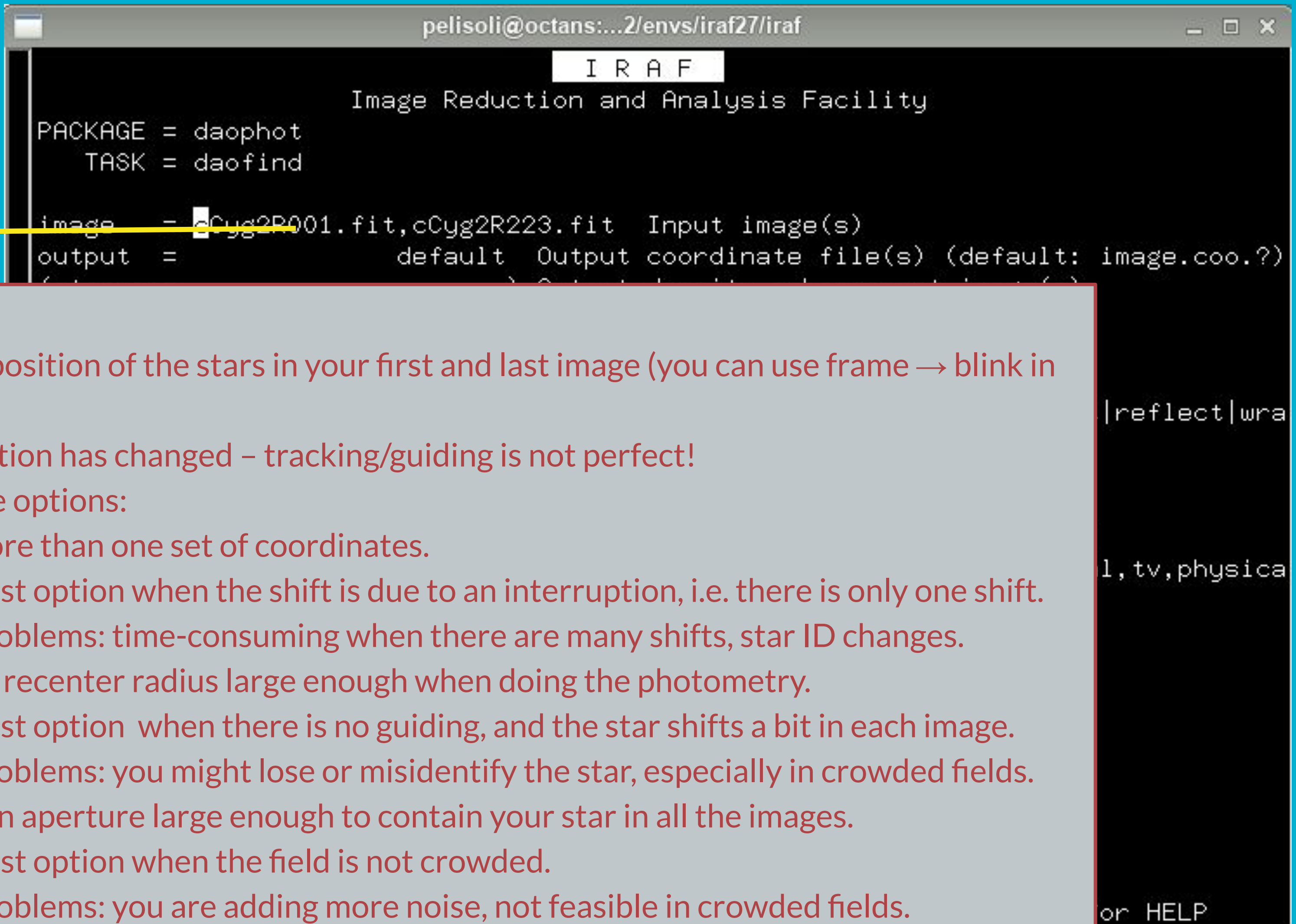
  sky = 415.
  sigma = 20.4

- Check the FWHM in the three images. We will use this to set the aperture and the sky region for the photometry.

DAOFIND

In which image to run it

DAOFIND

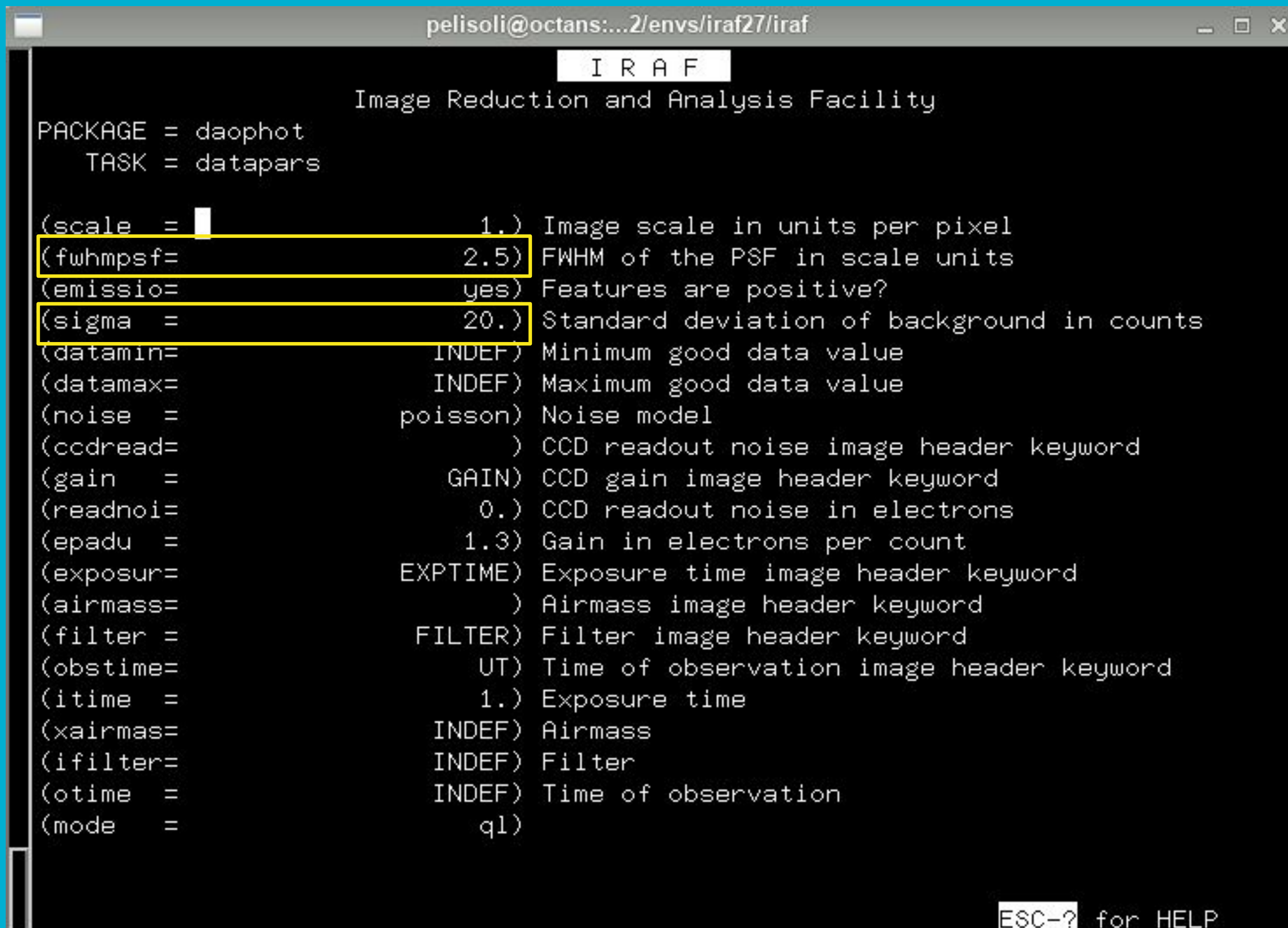


In which image to run it

WARNING!

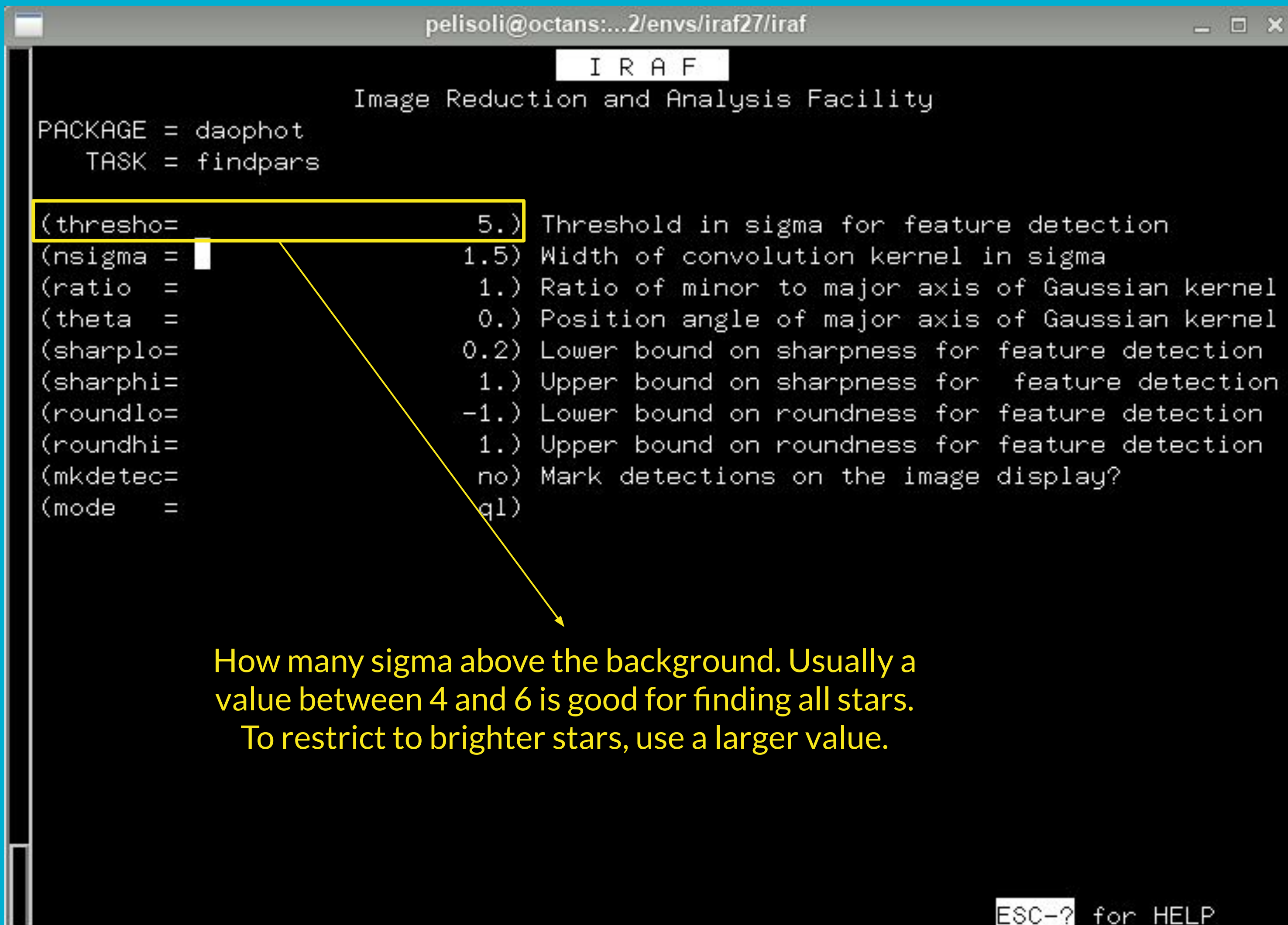
- Compare the position of the stars in your first and last image (you can use frame → blink in ds9).
- Likely the position has changed – tracking/guiding is not perfect!
- You have three options:
  - Have more than one set of coordinates.
    - Best option when the shift is due to an interruption, i.e. there is only one shift.
    - Problems: time-consuming when there are many shifts, star ID changes.
  - Define a recenter radius large enough when doing the photometry.
    - Best option when there is no guiding, and the star shifts a bit in each image.
    - Problems: you might lose or misidentify the star, especially in crowded fields.
  - Define an aperture large enough to contain your star in all the images.
    - Best option when the field is not crowded.
    - Problems: you are adding more noise, not feasible in crowded fields.

DAOFIND



pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = datapars
(scale = 1.) Image scale in units per pixel
(fwhmpsf= 2.5) FWHM of the PSF in scale units
(emissio= yes) Features are positive?
(sigma = 20.) Standard deviation of background in counts
(datamin= INDEF) Minimum good data value
(datamax= INDEF) Maximum good data value
(noise = poisson) Noise model
(ccdread=) CCD readout noise image header keyword
(gain = GAIN) CCD gain image header keyword
(readnoi= 0.) CCD readout noise in electrons
(epadu = 1.3) Gain in electrons per count
(exposur= EXPTIME) Exposure time image header keyword
(airmass=) Airmass image header keyword
(filter = FILTER) Filter image header keyword
(obstime= UT) Time of observation image header keyword
(itime = 1.) Exposure time
(xairmas= INDEF) Airmass
(ifilter= INDEF) Filter
(otime = INDEF) Time of observation
(mode = ql)
ESC-? for HELP

DAOFIND


pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = findpars
(thresho= 5.) Threshold in sigma for feature detection
(nsigma = 1.5) Width of convolution kernel in sigma
(ratio = 1.) Ratio of minor to major axis of Gaussian kernel
(theta = 0.) Position angle of major axis of Gaussian kernel
(sharplo= 0.2) Lower bound on sharpness for feature detection
(sharphi= 1.) Upper bound on sharpness for feature detection
(roundlo= -1.) Lower bound on roundness for feature detection
(roundhi= 1.) Upper bound on roundness for feature detection
(mkdetec= no) Mark detections on the image display?
(mode = ql)
How many sigma above the background. Usually a value between 4 and 6 is good for finding all stars. To restrict to brighter stars, use a larger value.
ESC-? for HELP

DAOFIND

- To check the stars that have been found, let’s mark them on the image.
- First, dump the coordinates and the ID of the stars onto a file:

```
tdump cCyg2R001.fit.coo.1 columns=c1,c2,c7 > coordsR
```

- Yo
- Ed

For the computers in Ondrejov, you need to edit some parameters from the tdump task first!

```
pelisoli@merak: ~
                                  I R A F
                   Image Reduction and Analysis Facility
PACKAGE = nttools
   TASK = tdump

table   = cCyg2R003.fit.coo.1  name of table to dump
(cdfile =              columns) output file for column definitions
(pfile  =               header) output file for header parameters
(datafil=               STDOUT) output file for table data
(columns=                     ) list of columns to be dumped
(rows   =                    -) range of rows to print
(pwidth =                   -1) output page width
(mode   =                   al)
```

DAOFIND

- To check the stars that have been found, let’s mark them on the image.
- First, dump the coordinates and the ID of the stars onto a file:

```
tdump cCyg2R001.fit.coo.1 columns=c1,c2,c7 > coordsR
```

- You might need to check the name of the columns:

```
tprint [FILE].coo.1 | less
```

- Edit the parameters of the task `tvmark`

DAOFIND

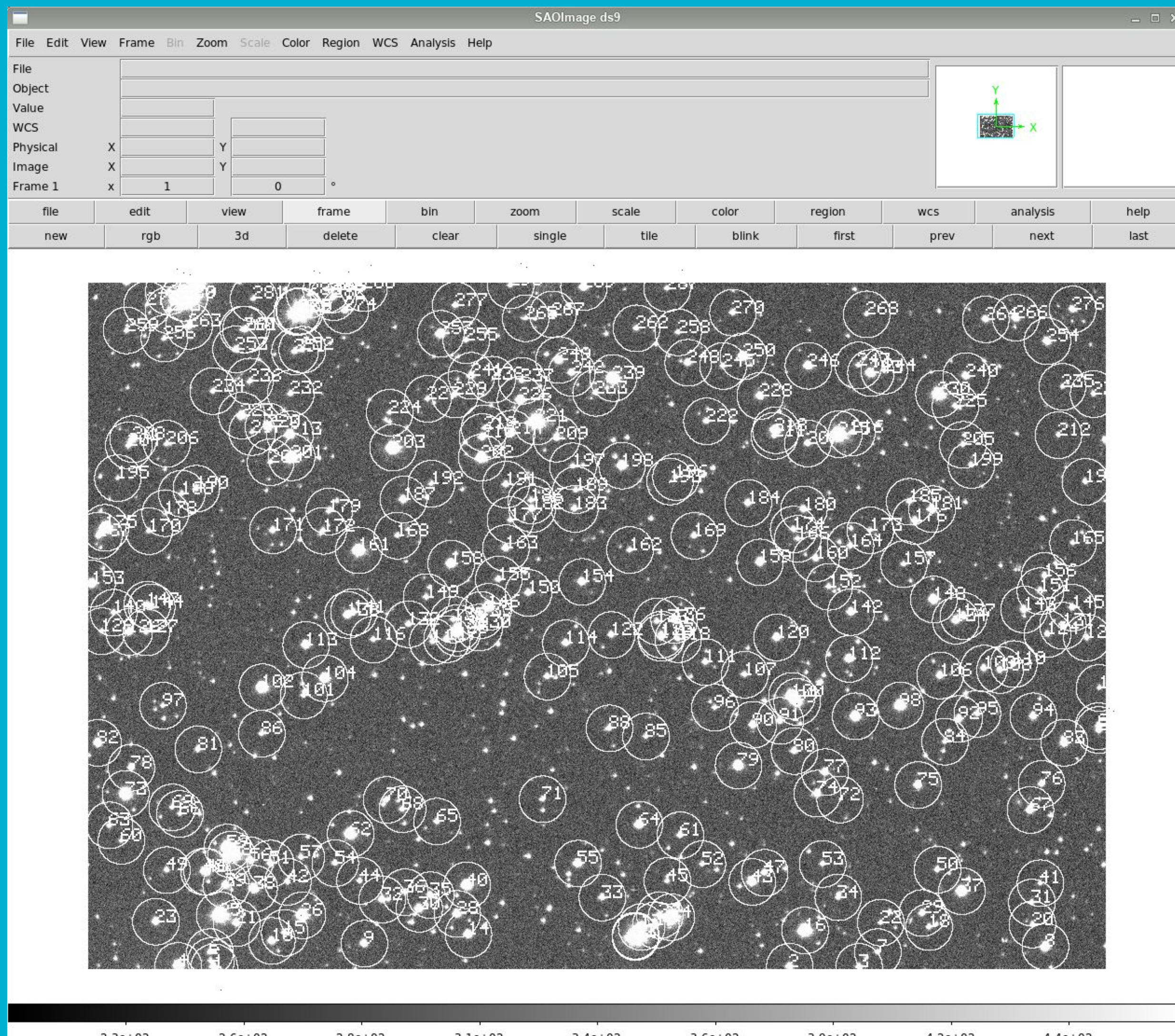
```
pelisoli@octans:...2/envs/iraf27/iraf

                                  I R A F
                    Image Reduction and Analysis Facility
PACKAGE = tv
   TASK = tvmark

frame   =                    1  Default frame number for display
coords  =              coordsR  Input coordinate list
(logfile=                     ) Output log file
(autolog=                   no) Automatically log each marking command
(outimag=                     ) Output snapped image
(deletio=                     ) Output coordinate deletions list
(command=                     ) Image cursor: [x y wcs] key [cmd]
(mark   =               circle) The mark type
(radii  =                   25) Radii in image pixels of concentric circles
(lengths=                    0) Lengths and width in image pixels of concentric
(font   =               raster) Default font
(color  =                    0) Gray level of marks to be drawn
(label  =                  yes) Label the marked coordinates
(number =                   no) Number the marked coordinates
(nxoffse=                    0) X offset in display pixels of number
(nyoffse=                    0) Y offset in display pixels of number
(pointsi=                    3) Size of mark type point in display pixels
(txsize =                    2) Size of text and numbers in font units
(toleran=                  1.5) Tolerance for deleting coordinates in image pixe
(interac=                   no) Mode of use
(mode   =                   ql)

ccdred>
```

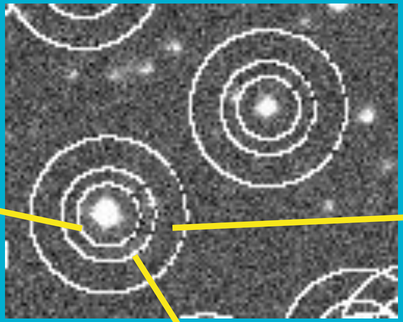
DAOFIND


SAOImage ds9
File Edit View Frame Bin Zoom Scale Color Region WCS Analysis Help
File
Object
Value
WCS
Physical X Y
Image X Y
Frame 1 x 1 0 °
file edit view frame bin zoom scale color region wcs analysis help
new rgb 3d delete clear single tile blink first prev next last
2.3e+02 2.6e+02 2.8e+02 3.1e+02 3.4e+02 3.6e+02 3.9e+02 4.2e+02 4.4e+02

Photometry

- `tvmark` is also useful to help us define the aperture, annulus, and dannulus

Aperture: where the flux of the star will be measured. Usually ~2.5 x FWHM



Dannulus: size of the ring to count the background. ~5-10 pixels

Annulus: distance at which to start counting the background. At least 2.5 x FHWM

* For a Gaussian distribution:
FWHM = 2.35σ
99.99% of the light is contained within
4σ = 1.7FWHM

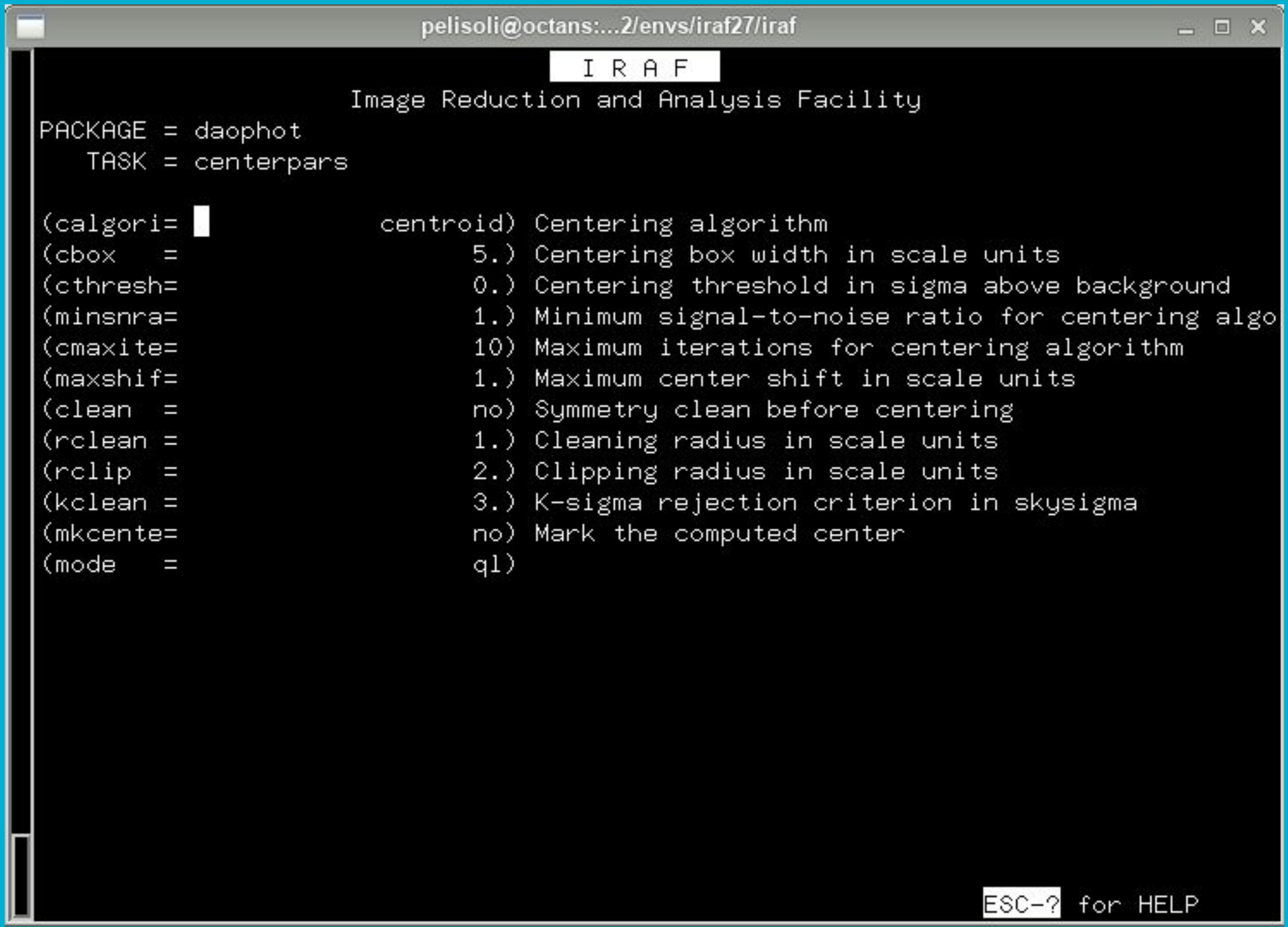
PHOT

List of reduced R (or V) images.

```
pelisoli@octans:...2/envs/iraf27/iraf
                                   I R A F
                     Image Reduction and Analysis Facility
PACKAGE = daophot
   TASK = phot

image   =           @cimgsR_beg  Input image(s)
coords  =   cCyg2R001.fit.coo.1  Input coordinate list(s) (default: image.coo.?)
output  =               default  Output photometry file(s) (default: image.mag.?)
skyfile =                        Input sky value file(s)
(plotfil=                     ) Output plot metacode file
(datapar=                     ) Data dependent parameters
(centerp=  :e                 ) Centering parameters
(fitskyp=  :e                 ) Sky fitting parameters
(photpar=  :e                 ) Photometry parameters
(interac=                   no) Interactive mode?
(radplot=                   no) Plot the radial profiles?
(icomman=                     ) Image cursor: [x y wcs] key [cmd]
(gcomman=                     ) Graphics cursor: [x y wcs] key [cmd]
(wcsin  =             )_.wcsin) The input coordinate system (logical,tv,physical
(wcsout =            )_.wcsout) The output coordinate system (logical,tv,physica
(cache  =             )_.cache) Cache the input image pixels in memory?
(verify =            )_.verify) Verify critical phot parameters?
(update =            )_.update) Update critical phot parameters?
(verbose=           )_.verbose) Print phot messages?
(graphic=          )_.graphics) Graphics device
(display=           )_.display) Display device
More
                                                          ESC-? for HELP
```

PHOT


pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = centerpars
(calgori= centroid) Centering algorithm
(cbox = 5.) Centering box width in scale units
(cthresh= 0.) Centering threshold in sigma above background
(minsnra= 1.) Minimum signal-to-noise ratio for centering algo
(cmaxite= 10) Maximum iterations for centering algorithm
(maxshif= 1.) Maximum center shift in scale units
(clean = no) Symmetry clean before centering
(rclean = 1.) Cleaning radius in scale units
(rclip = 2.) Clipping radius in scale units
(kclean = 3.) K-sigma rejection criterion in skysigma
(mkcente= no) Mark the computed center
(mode = ql)
ESC-? for HELP

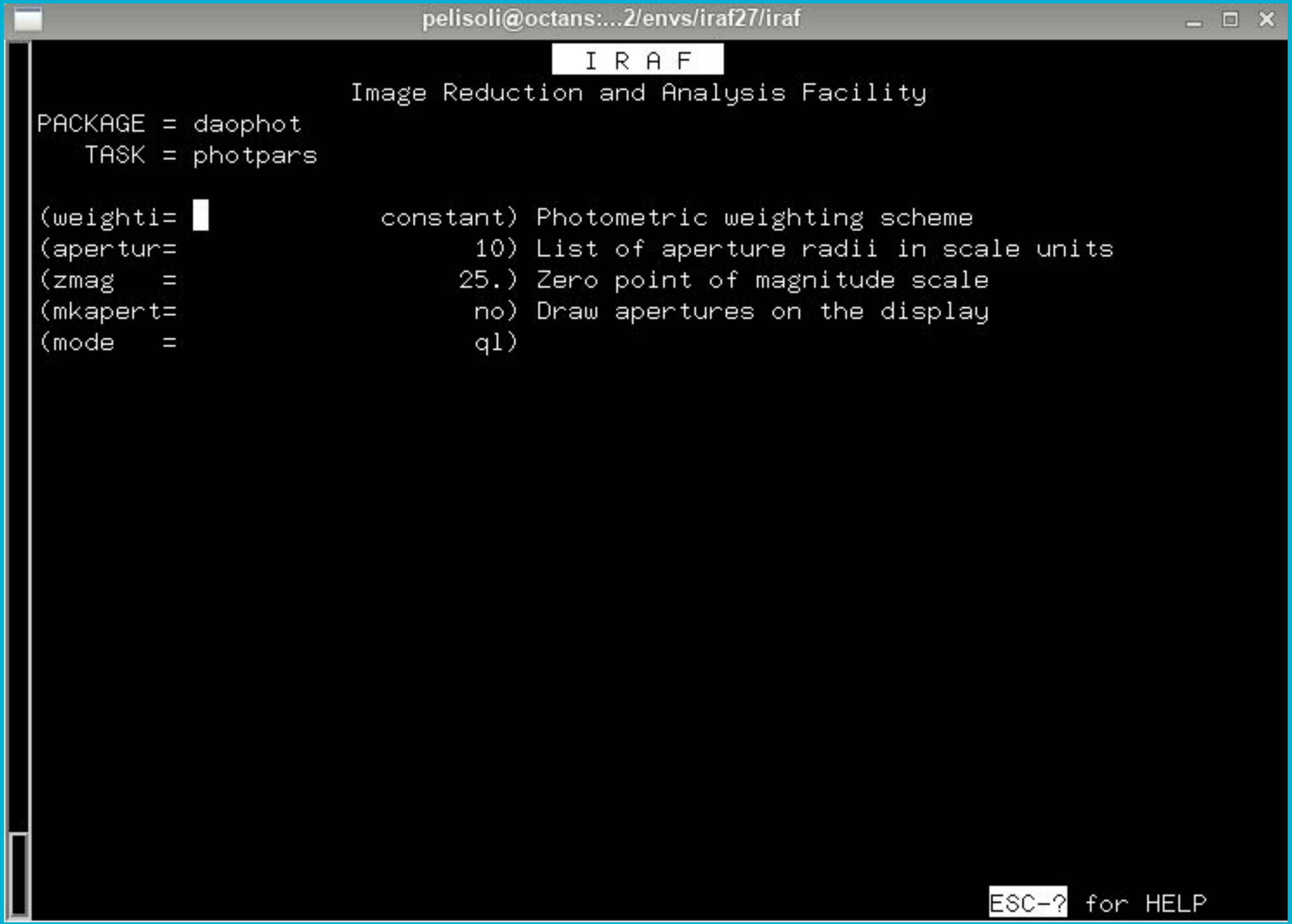
PHOT

```
pelisoli@octans:...2/envs/iraf27/iraf
                                    I R A F
                    Image Reduction and Analysis Facility
PACKAGE = daophot
   TASK = fitskypars

(salgori=                     mode) Sky fitting algorithm
(annulus=                      15.) Inner radius of sky annulus in scale units
(dannulu=                      10.) Width of sky annulus in scale units
(skyvalu=                     415.) User sky value
(smaxite=                       20) Maximum number of sky fitting iterations
(sloclip=                       0.) Lower clipping factor in percent
(shiclip=                       0.) Upper clipping factor in percent
(snrejec=                       50) Maximum number of sky fitting rejection iteratio
(sloreje=                       3.) Lower K-sigma rejection limit in sky sigma
(shireje=                       3.) Upper K-sigma rejection limit in sky sigma
(khist  =                       3.) Half width of histogram in sky sigma
(binsize=                      0.1) Binsize of histogram in sky sigma
(smooth =                       no) Boxcar smooth the histogram
(rgrow  =                       0.) Region growing radius in scale units
(mksky  =                       no) Mark sky annuli on the display
(mode   =                       ql)

                                                          ESC-? for HELP
```

PHOT


pelisoli@octans:...2/envs/iraf27/iraf
I R A F
Image Reduction and Analysis Facility
PACKAGE = daophot
TASK = photpars
(weighti= constant) Photometric weighting scheme
(apertur= 10) List of aperture radii in scale units
(zmag = 25.) Zero point of magnitude scale
(mkapert= no) Draw apertures on the display
(mode = ql)
ESC-? for HELP

Photometry

- Dump the photometry into a text file:

```
ls *R*mag.1 > Rmag_files
tdump @Rmag_files columns=c4,c7,c8,c29,c30,c31 > R_mags
```

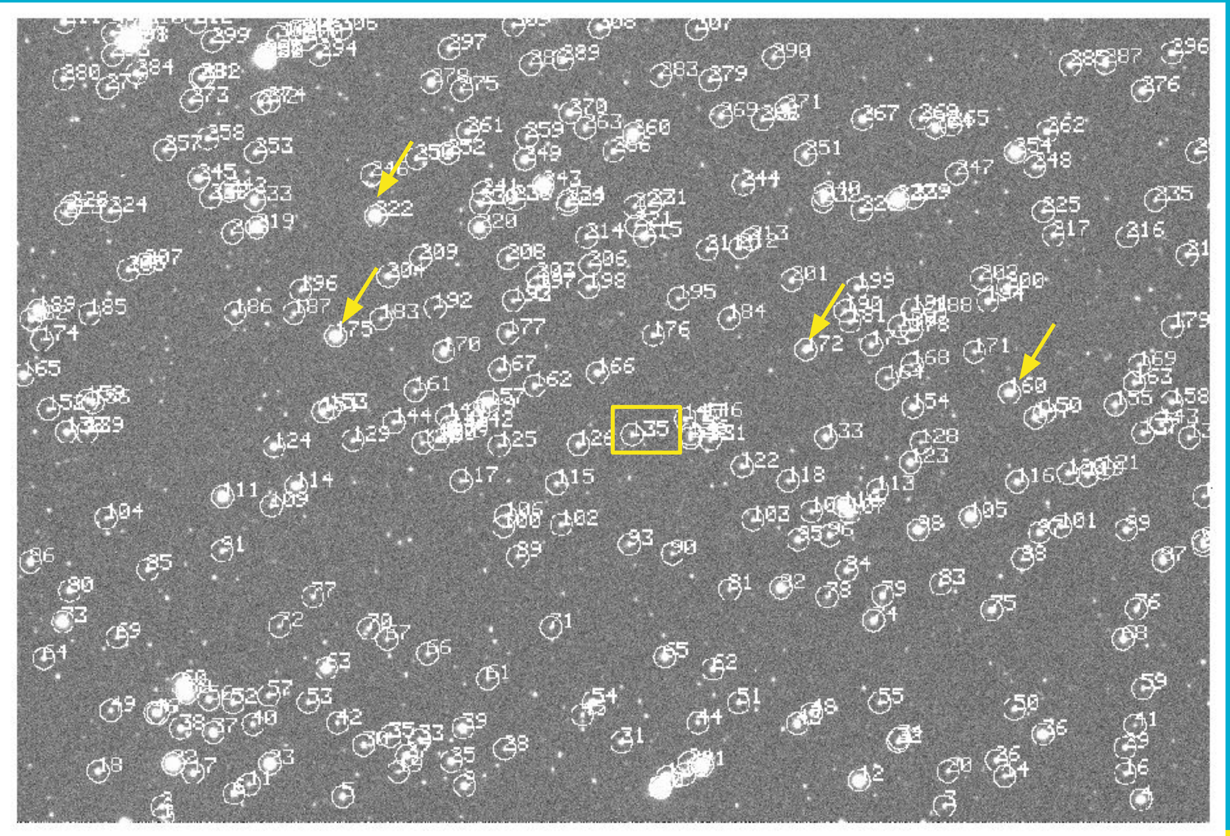
* if `tdump` refuses to read from a list, use awk to create a file with one tdump per line:

```
ls *mag.1 | awk '{printf "tdump %s columns=c4,c7,c8,c29,c30,c31 >> R_mags\n", $1}' > get_mags
cl < get_mags
```

- c4 = star ID, c7 = x coordinate, c8 = y coordinate, c29 = magnitude, c30 = magnitude error, c31= flux.

Photometry

- Check the ID of your star and of a few comparison stars with `tvmark`. https://aladin.u-strasbg.fr/AladinLite/ might be useful to help identify your star.
- Comparison stars are needed to remove background variations from the light curve.



Photometry

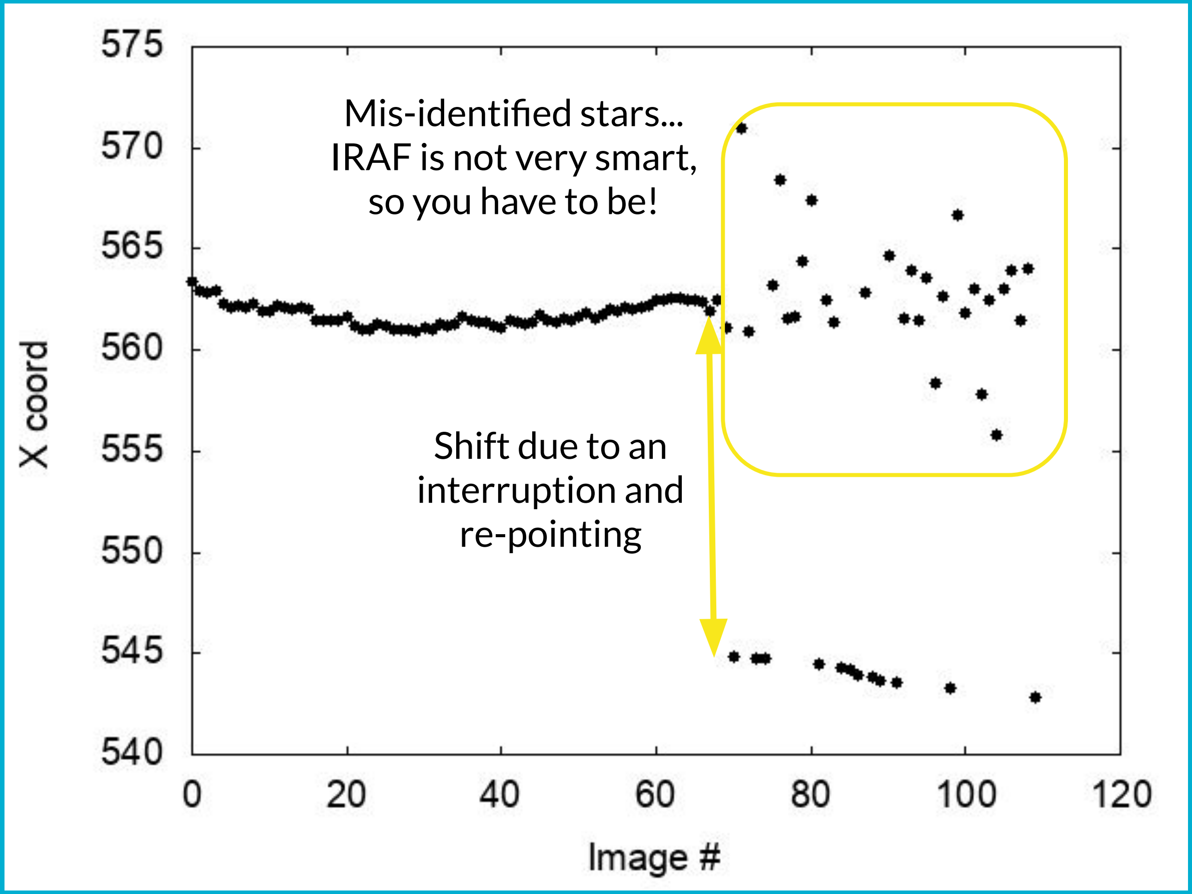
- Copy the photometry of the star and each comparison into separate files.

```
awk '{if ($1==135) print;}' R_mags > R_star
awk '{if ($1==160) print;}' R_mags > R_comp1
awk '{if ($1==172) print;}' R_mags > R_comp2
awk '{if ($1==175) print;}' R_mags > R_comp3
awk '{if ($1==222) print;}' R_mags > R_comp4
```

↳ Replace with correct IDs

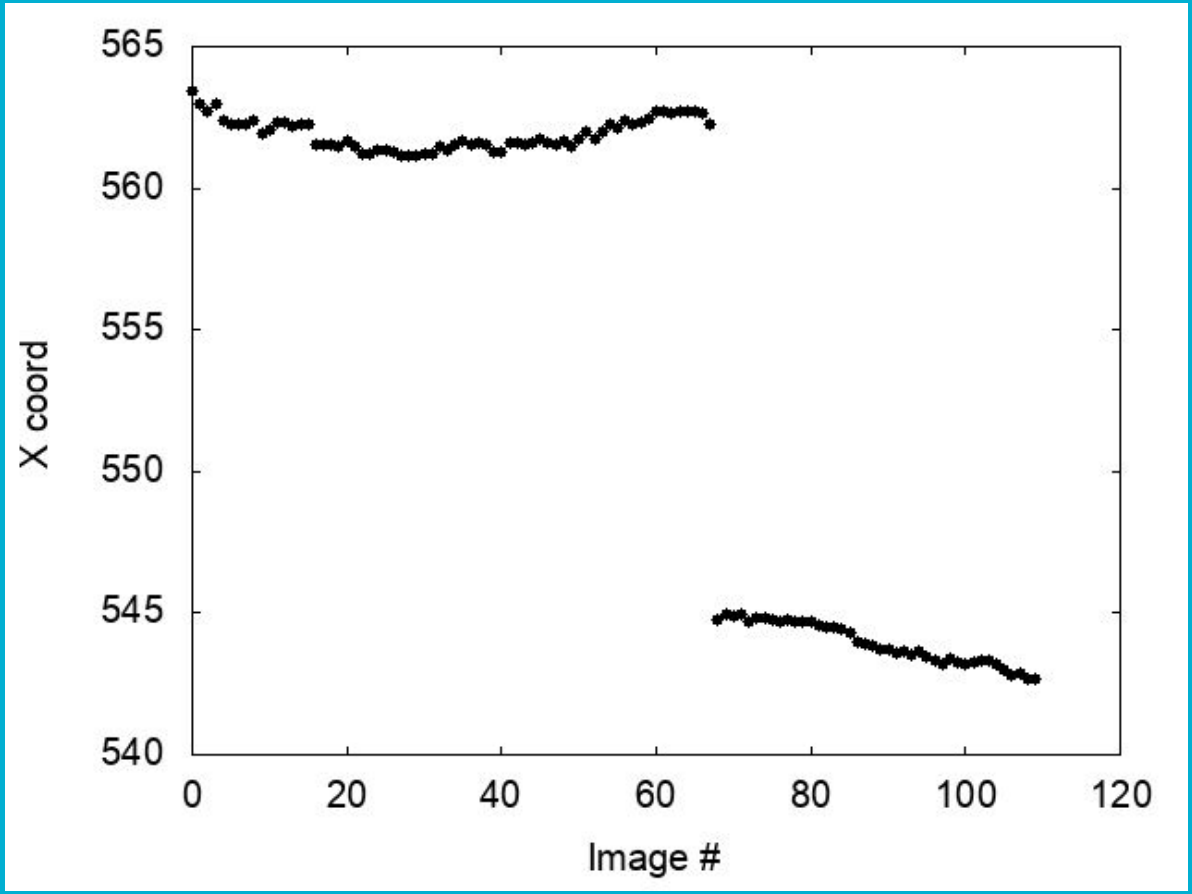
- It is a good sanity check to plot the x and y coordinates of each star, to make sure it was correctly identified in all images.

Photometry



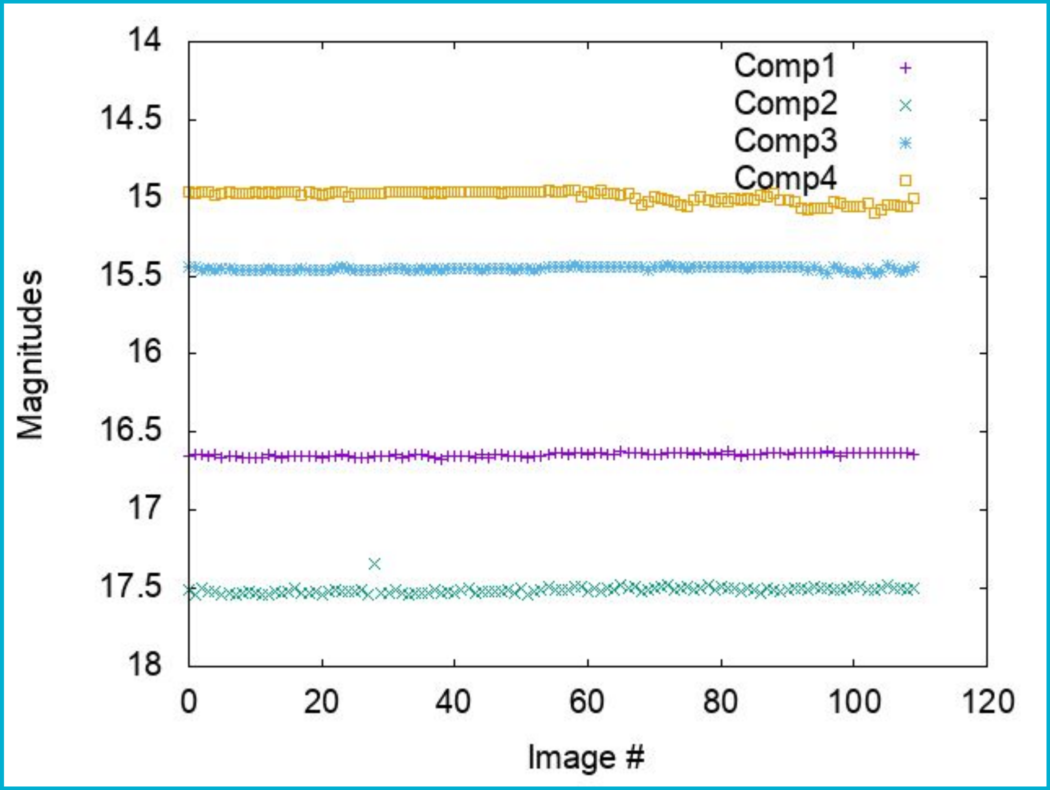
575
570
565
560
555
550
545
540
0
20
40
60
80
100
120
X coord
Image #
Mis-identified stars...
IRAF is not very smart,
so you have to be!
Shift due to an
interruption and
re-pointing

Photometry



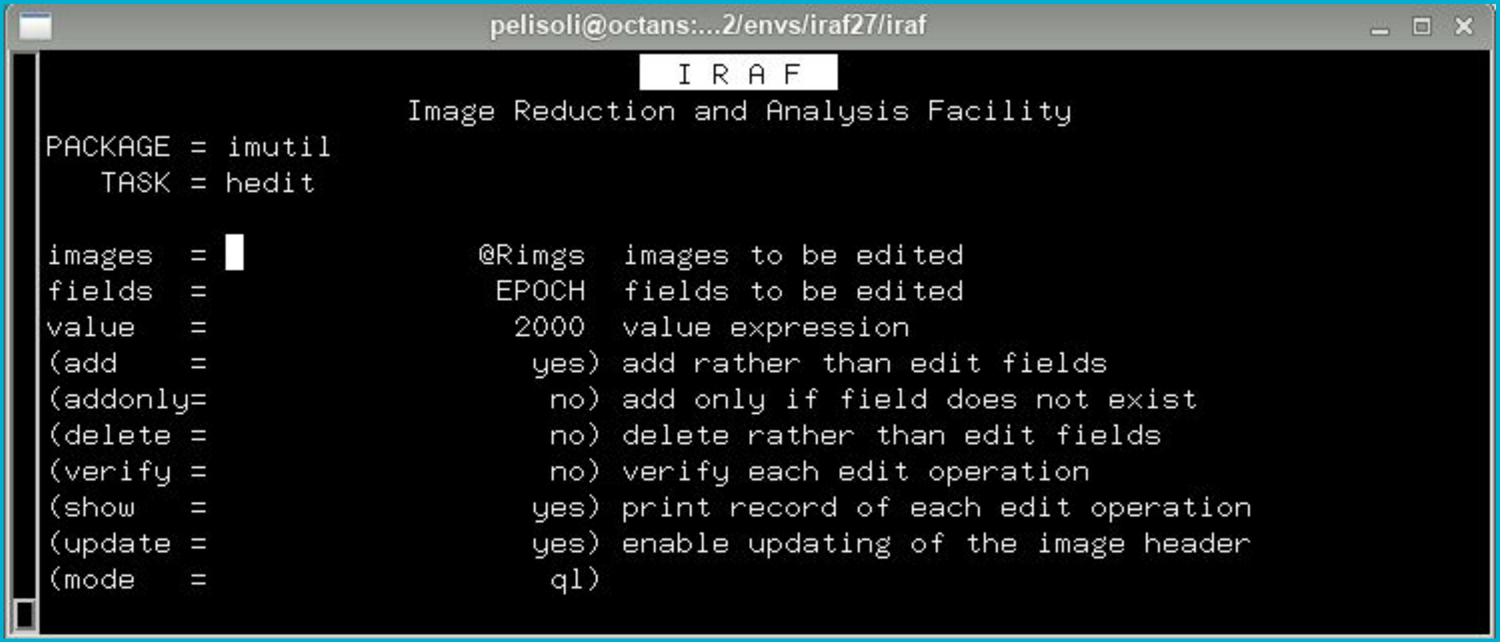
Photometry

- Another good check is to plot the magnitudes of your comparison stars. They have to be fairly constant!



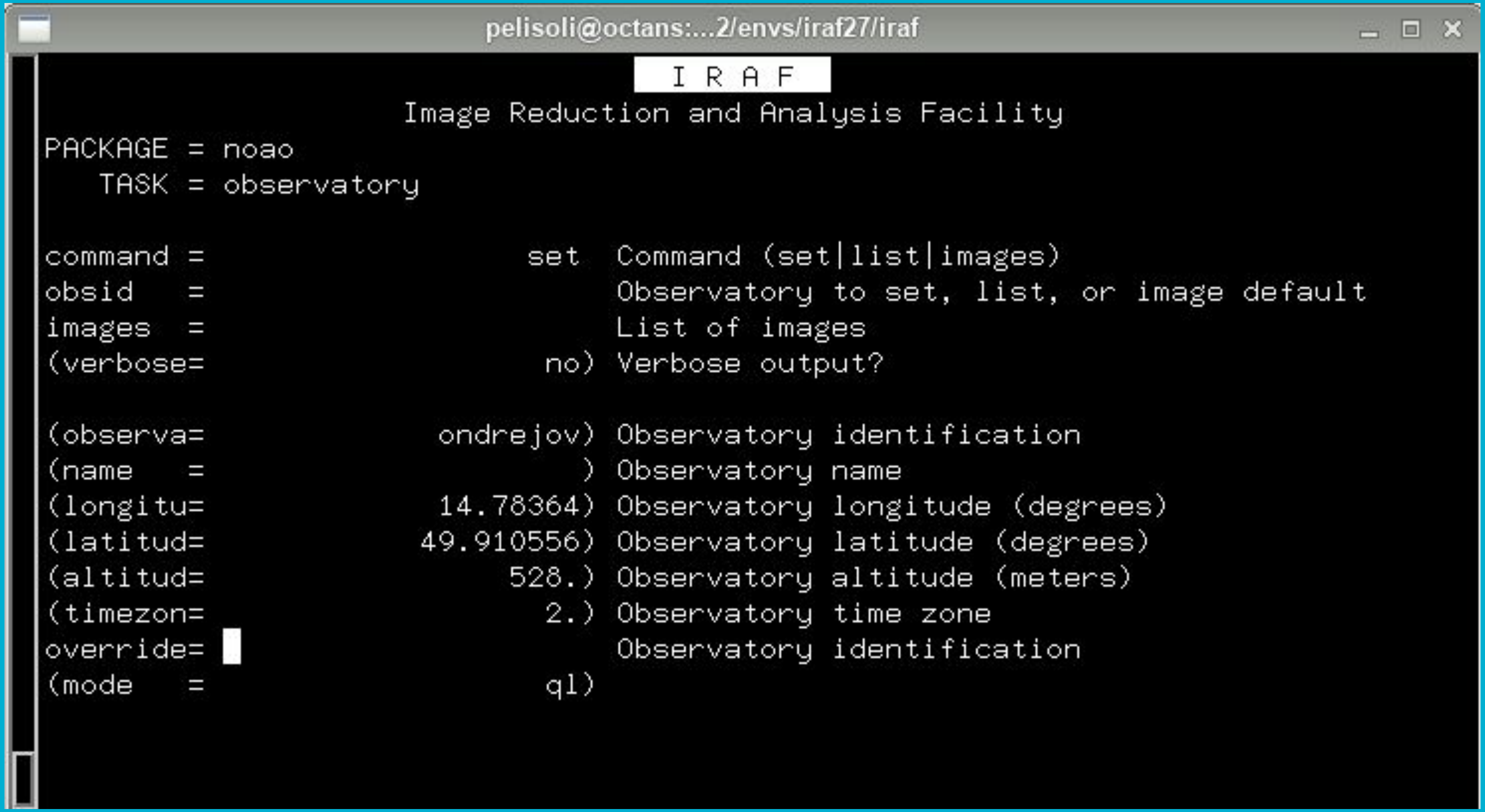
Building the light curve

- To turn our measurements into a light curve, we need the times for each observation. We will use the task `setjd` to obtain that.
- The headers of our images are missing one important information: coordinates (RA, DEC, Epoch). Use the task `hedit` to add those to all images.



Building the light curve

- We also need to set the observatory parameters to be used for `setjd`. We do that with the task observatory:



Building the light curve

```
pelisoli@octans:...2/envs/iraf27/iraf
                                I R A F
                  Image Reduction and Analysis Facility
PACKAGE = onedspec
   TASK = setjd

images  =                   @Rimgs  Images
(observa=                 obspars) Observatory of observation
(date   =                date-obs) Date of observation keyword
(time   =                      ut) Time of observation keyword
(exposur=                 exptime) Exposure time keyword
(ra     =                      ra) Right ascension (hours) keyword
(dec    =                     dec) Declination (degrees) keyword
(epoch  =                   epoch) Epoch (years) keyword

(jd     =                      jd) Output Julian date keyword
(hjd    =                     hjd) Output Helocentric Julian date keyword
(ljd    =                     ljd) Output local Julian date keyword

(utdate =                     yes) Is observation date UT?
(uttime =                     yes) Is observation time UT?
(listonl=                      no) List only without modifying images?
(mode   =                      ql)

                                                          ESC-? for HELP
```

`setjd > R_jd`

Building the light curve

- To do differential photometry, we need to normalise the magnitudes of the star and of the comparison stars. First, check what is the average magnitude:

```
awk '{sum+=$5;n++} END {print sum/n;}' R_star
```

- Then subtract it from each value: → replace with calculated average

```
awk '{printf "%7.4f %6.4f\n", $5-18.7529, $6}' R_star > mag_star
```

- Repeat that for all the comparison stars, and combine them into one file:

```
paste mag_comp1 mag_comp2 mag_comp3 mag_comp4 > all_comp
```

- Average the comparison stars:

```
awk '{printf "%7.4f %6.4f\n", ($1+$3+$5+$7)/4.0,
sqrt($2*$2+$4*$4+$6*$6+$8*$8)}' all_comp > mag_comp
```

Building the light curve

- Combine the magnitudes of the star and the comparison magnitude:

```
paste mag_star mag_comp > comb_mag
```

- Subtract the comparison from the star to remove background variations:

```
awk '{printf "%7.4f %6.4f\n", ($1-$3), sqrt($2*$2+$4*$4)}'
comb_mag > diff_mag
```

- Select the column containing the Heliocentric Julian Date from the file created with setjd:

```
awk '!/#/ {print $3}' R_jd > R_hjd
```

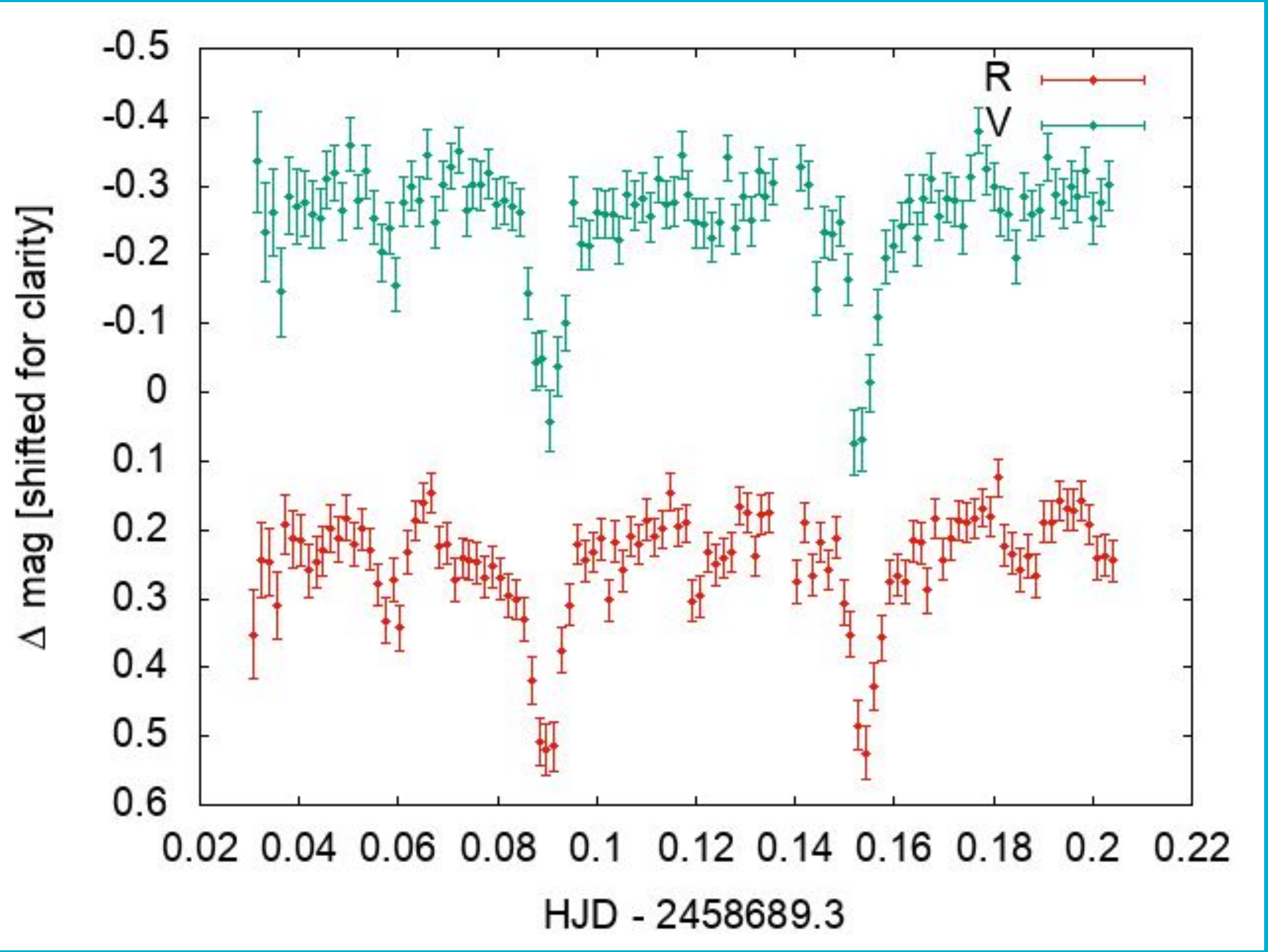
- Combine that with the magnitude to obtain the lightcurve:

```
paste R_hjd diff_mag > R_lightcurve
```

Voilà! Now you have a light curve.

Repeat the same for the other filter.

Light curves



Photometry – summary

- Create master files for bias, flat, and dark (`zerocombine`, `flatcombine`, `darkcombine`).
- Reduce the science images using `ccdproc`.
- Measure sky and FWHM with `imexamine`.
- Use the dask `daofind` to find the stars; do not forget to change the `datapars` according to your measurements, and set the threshold in `findpars`.
- Use the task `phot` to do the photometry; do not forget to update `centerpars`, `fitskypars` and `photpars`.
- Check ID for your star and comparison stars using `display` and `tvmark`.
- Inspect the coordinates for the star and comparison stars to guarantee there was no misidentification.
- Inspect the magnitudes of the comparison stars; they should be fairly constant.
- Use `observatory` and `setjd` to obtain the times of observation.
- Paste the times and differential magnitude (star - averaged comparison) into one file to obtain the light curve.

Optional task

- We did a lot by hand, but the commands can be combined onto a script to make the process more automatic!
- If you are familiar with coding (shell or python are the more adequate in this case), you could try to write a script.