

Advanced Astronomical Observations 2021

Session 22: World Coordinate System (WCS)

Kinoshita Daisuke

09 June 2021
publicly accessible version

About this file...

- Important information about this file
 - The author of this file is Kinoshita Daisuke.
 - The original version of this file was used for the course “Advanced Astronomical Observations” (course ID: AS6005) offered at Institute of Astronomy, National Central University from February 2021 to June 2021.
 - The file is provided in the hope that it will be useful, but there is no guarantee for the correctness. Use this file at your own risk.
 - If you are willing to use this file for your study, please feel free to use. I’ll be very happy to receive feedback from you.
 - If you are willing to use this file for your teaching, please contact to Kinoshita Daisuke. When you use this file partly or entirely, please mention clearly that the author of the original version is Kinoshita Daisuke. Please help me to improve the contents of this file by sending your feedback.
 - Contact address: <https://www.instagram.com/daisuke23888/>

For this session, we try to add WCS-related keywords to FITS files.

1 Python scripts for this session

All the Python scripts needed for this session are available at github.com. Visit the web page https://github.com/kinoshitadaisuke/ncu_advobs_202102 to view files. (Fig. 1)

You may try following command to download all the Python scripts for this session.

```
% git clone https://github.com/kinoshitadaisuke/ncu_advobs_202102.git
Cloning into 'ncu_advobs_202102'...
remote: Enumerating objects: 296, done.
remote: Counting objects: 100% (94/94), done.
remote: Compressing objects: 100% (90/90), done.
remote: Total 296 (delta 42), reused 0 (delta 0), pack-reused 202
Receiving objects: 100% (296/296), 5.40 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (150/150), done.
% ls
ncu_advobs_202102/
% ls -l ncu_advobs_202102/
total 1
-rw-r--r--  1 daisuke  wheel  35149 Jun  8 23:15 LICENSE
-rw-r--r--  1 daisuke  wheel   568 Jun  8 23:15 README
-rw-r--r--  1 daisuke  wheel   687 Jun  8 23:15 REQUIREMENTS
-rw-r--r--  1 daisuke  wheel   342 Jun  8 23:15 download_python.csh
-rw-r--r--  1 daisuke  wheel   495 Jun  8 23:15 download_stds.csh
drwxr-xr-x  2 daisuke  wheel   512 Jun  8 23:15 s16_skybackground/
```

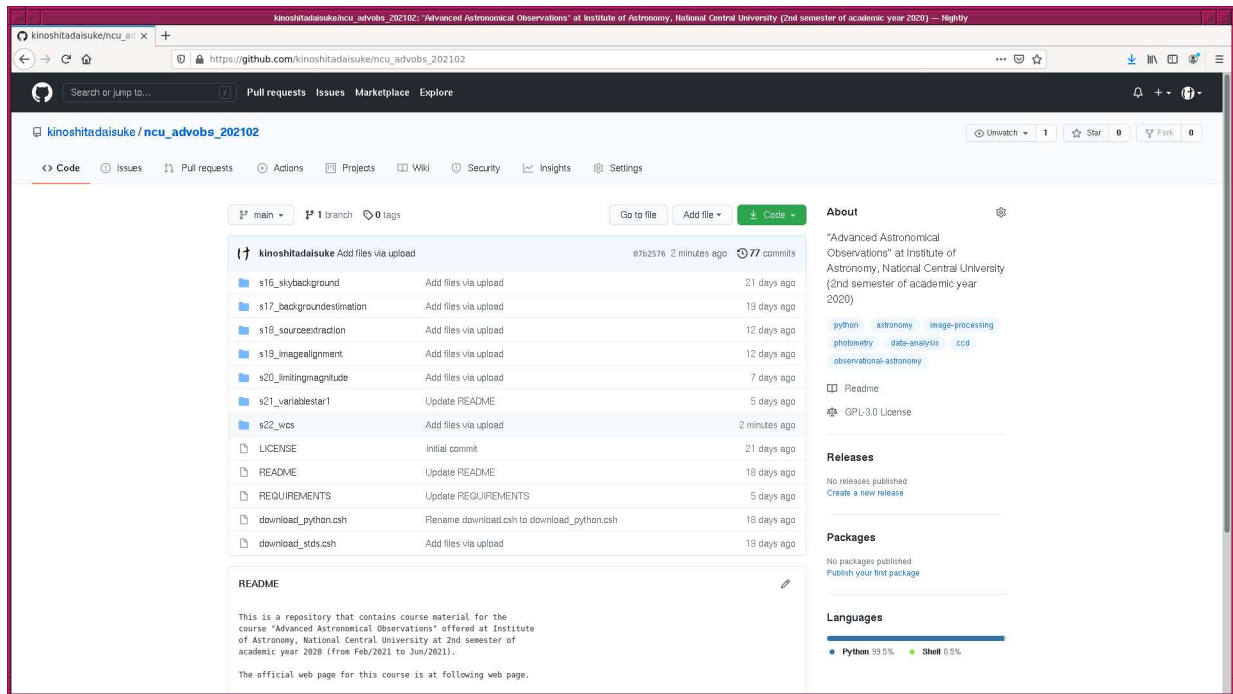


Figure 1: The GitHub repository for Python scripts for this course.

```

drwxr-xr-x  2 daisuke  wheel   512 Jun  8 23:15 s17_backgroundestimation/
drwxr-xr-x  2 daisuke  wheel   512 Jun  8 23:15 s18_sourceextraction/
drwxr-xr-x  2 daisuke  wheel   512 Jun  8 23:15 s19_imagealignment/
drwxr-xr-x  2 daisuke  wheel   512 Jun  8 23:15 s20_limitingmagnitude/
drwxr-xr-x  2 daisuke  wheel   512 Jun  8 23:15 s21_variablestar1/
drwxr-xr-x  2 daisuke  wheel   512 Jun  8 23:15 s22_wcs/
% ls -l ncu_advobs_202102/s22_wcs/
total 3
-rw-r--r--  1 daisuke  wheel    60 Jun  8 23:15 README
-rw-r--r--  1 daisuke  wheel  1440 Jun  8 23:15 ao2021_s22_01.py
-rw-r--r--  1 daisuke  wheel  2157 Jun  8 23:15 ao2021_s22_02.py
-rw-r--r--  1 daisuke  wheel  2667 Jun  8 23:15 ao2021_s22_03.py
-rw-r--r--  1 daisuke  wheel  2710 Jun  8 23:15 ao2021_s22_04.py
-rw-r--r--  1 daisuke  wheel  1189 Jun  8 23:15 ao2021_s22_05.py
-rw-r--r--  1 daisuke  wheel  1111 Jun  8 23:15 ao2021_s22_06.py
-rw-r--r--  1 daisuke  wheel  2365 Jun  8 23:15 ao2021_s22_07.py
-rw-r--r--  1 daisuke  wheel  5242 Jun  8 23:15 ao2021_s22_08.py
-rw-r--r--  1 daisuke  wheel  5074 Jun  8 23:15 ao2021_s22_09.py
-rw-r--r--  1 daisuke  wheel 2249158 Jun  8 23:15 ginga_0085_dfkw.png
% head -15 ncu_advobs_202102/s22_wcs/ao2021_s22_01.py
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2021/06/08 17:30:07 (CST) daisuke>
#

# importing astropy module
import astropy.coordinates
import astropy.units

# importing astroquery module
import astroquery.gai

```

```
# importing ssl module
import ssl
```

If above command fails on your computer, check whether or not you have the command `git`. You see something like following if you have the command `git` on your computer.

```
% which git
/usr/pkg/bin/git
```

You see following if you do not have the command `git` installed on your computer.

```
% which git
git: Command not found.
```

If you do not have the command `git`, install `git` on your computer.

2 Installation of WCSTools

2.1 Official website of WCSTools

Visit the official website of WCSTools (Fig. 2) to learn about it.

- WCSTools
 - <http://tdc-www.harvard.edu/wcstools/>

The screenshot shows the official website of WCSTools. The page is titled "WCSTools: Image World Coordinate System Utilities" and is hosted on "tdc-www.harvard.edu/wcstools/". The website includes a search bar for the "Telescope Data Center" and a description of the software. The main content is organized into sections for "WCSTools Capabilities" and "WCSTools Programs".

WCSTools Capabilities:

- Handles all [FITS](#) image data types: 8-bit unsigned integer, 16- and 32-bit signed integer, 32- and 64-bit IEEE floating point, plus non-standard 16-bit unsigned integer (BITPIX=-16).
- Accesses IRAF .imh (versions 1 and 2) files as well as FITS files
- Uses any of several reference catalog formats from CDROMs or identically-structured online files,
- Uses Mark Calabretta's [WCSLIB library](#), and the [SAO WCSTools library](#), which includes [additional capabilities](#)
- **Is Y2K-compliant.**
- **Implements images with more than one WCS**
- Uses same WCS subroutines as [SAOimage](#), [SAOtngr](#), [ds9](#), and [skycat](#) image browsing programs.
- [Can be installed as an IRAF package](#)

WCSTools Programs:

- [Image WCS](#) ([imwcs](#)) ([remap](#)) ([sky2xy](#)) ([xy2sky](#)) ([SAOimage](#)) [[Getting Good Coordinates](#)]
- [Catalogs and Image WCS](#) ([imcat](#)) ([imstar](#)) [[Environment Variables](#)]
- [Image Header Utilities](#) ([cpead](#)) ([delhead](#)) ([edhead](#)) ([gethead](#)) ([imhead](#)) ([imsize](#)) ([keyhead](#)) ([sethead](#))
- [Image Utilities](#) ([getfits](#)) ([getpix](#)) ([i2f](#)) ([newfits](#)) ([setpix](#)) ([sumpix](#))
- [Catalog Utilities](#) ([imcat](#)) ([scat](#)) ([skycat](#))

At the bottom of the page, there are links to [USNO-B1.0 Catalog](#) and [2MASS Point Source Catalog](#).

Figure 2: The official website of WCSTools.

2.2 Downloading WCSTools

Download source code of WCSTools.

```
% curl -o wcstools-3.9.6.tar.gz \
? http://tdc-www.harvard.edu/software/wcstools/wcstools-3.9.6.tar.gz
  % Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
                             Dload  Upload   Total   Spent    Left   Speed
100 8412k  100 8412k    0     0   550k      0  0:00:15  0:00:15  --:--:--  531k
% ls -l
total 9
-rw-r--r--  1 daisuke  taiwan  8614498 Jun  8 09:21 wcstools-3.9.6.tar.gz
```

2.3 Extracting files

Extract files of WCSTools using the command tar.

```
% tar xzvf wcstools-3.9.6.tar.gz
x wcstools-3.9.6/
x wcstools-3.9.6/libwcs/
x wcstools-3.9.6/libwcs/Makefile
x wcstools-3.9.6/libwcs/Files
x wcstools-3.9.6/libwcs/NEWS
x wcstools-3.9.6/libwcs/Readme
.....
x wcstools-3.9.6/isdate.c
x wcstools-3.9.6/isfile.c
x wcstools-3.9.6/isfits.c
x wcstools-3.9.6/isimlist.c
x wcstools-3.9.6/isnum.c
% ls -l
total 9
drwxr-xr-x  7 daisuke  taiwan    2560 Sep  1  2020 wcstools-3.9.6/
-rw-r--r--  1 daisuke  taiwan  8614498 Jun  8 09:21 wcstools-3.9.6.tar.gz
% cd wcstools-3.9.6
% ls
COPYING          fdump.par        immatch.c        setpix.c
Install          filedir.c        imresize.c       simpos.c
Linkbins*       filename.c       imrot.c          simposx.c
MakeLinks*      fileroot.c       imsize.c         sky2xy.c
Makefile        filext.c         imsmooth.c       skycoor.c
.....
crlf.c          imcat.c          revup*           xy2sky1.c
delhead.c       imcat0.c         revup.sed        yb6cat.c
delwcs.c        imextract.c      scat.c           zgethead*
edhead.c        imfill.c         sethead.c        zgethead1*
fdump.list      imhead.c         setheadx.c       zimhead*
```

2.4 Compiling WCSTools

Compile WCSTools.

```

% make all
cd libwcs; make
make[1]: Entering directory '/home/daisuke/tex/astro/NCU/Lecture/AdvObs_2020b/22
_wcs/script_22/wcstools-3.9.6/libwcs'
cc -g -D_FILE_OFFSET_BITS=64 -I/home/daisuke/include -I/usr/local/include -I/usr
/pkg/include -I/usr/pkg/xorg/include -I/usr/X11R7/include -I/usr/include -I/usr/
X11R6/include -I/usr/include -c -o imsetwcs.o imsetwcs.c
cc -g -D_FILE_OFFSET_BITS=64 -I/home/daisuke/include -I/usr/local/include -I/usr
/pkg/include -I/usr/pkg/xorg/include -I/usr/X11R7/include -I/usr/include -I/usr/
X11R6/include -I/usr/include -c -o imgetwcs.o imgetwcs.c

.....

cc -g -D_FILE_OFFSET_BITS=64 -o bin/isrange isrange.c
cc -g -D_FILE_OFFSET_BITS=64 -o bin/isfits isfits.c libwcs/libwcs.a -lm
cc -g -D_FILE_OFFSET_BITS=64 -o bin/isfile isfile.c libwcs/libwcs.a -lm
cc -g -D_FILE_OFFSET_BITS=64 -o bin/simpos simpos.c libwcs/libwcs.a -lm
cc -g -D_FILE_OFFSET_BITS=64 -o bin/bincat bincat.c libwcs/libwcs.a -lm
% ls bin
addpix*    edhead*    gethead*    imhead*    imwcs*    scat*      sumpix*
bincat*    filename*  getpix*     immatch*   isfile*   sethead*   wcshead*
char2sp*   fileroot*  gettab*     imresize*  isfits*   setpix*    xy2sky*
conpix*    filext*    htmlsp*     imrot*     isnum*    simpos*
cphead*    fixpix*    i2f*        imsize*    isrange*  sky2xy*
crlf*      getcol*    imcat*      imsmooth*  keyhead*  skycoor*
delhead*   getdate*   imextract*  imstack*   newfits*  sp2char*
delwcs*    getfits*   imfill*     imstar*    remap*    subpix*

```

2.5 Installation of WCSTools

Install WCSTools.

```

% su
Password:
# mkdir -p /usr/local/bin
# cp -pi bin/* /usr/local/bin/
# ls /usr/local/bin/
addpix*    edhead*    gethead*    imhead*    imwcs*    scat*      sumpix*
bincat*    filename*  getpix*     immatch*   isfile*   sethead*   wcshead*
char2sp*   fileroot*  gettab*     imresize*  isfits*   setpix*    xy2sky*
conpix*    filext*    htmlsp*     imrot*     isnum*    simpos*
cphead*    fixpix*    i2f*        imsize*    isrange*  sky2xy*
crlf*      getcol*    imcat*      imsmooth*  keyhead*  skycoor*
delhead*   getdate*   imextract*  imstack*   newfits*  sp2char*
delwcs*    getfits*   imfill*     imstar*    remap*    subpix*
# cp -pi MakeLinks /tmp/
# cd /usr/local/bin/
# sh /tmp/MakeLinks
# ls
addpix*    gettab*    immsao@    imsmooth*  imwiras@  keyhead*   ssdss@
bincat*    htmlsp*    immsdss@  imstack*   imwppm@   nedpos@    stmc@
cdspos@    i2f*       immtmc@   imstar*    imwsao@   newfits*   stmx@
char2sp*   imact@     immtmx@   imtmc@     imwsdss@  remap*     sty2@
conpix*    imcat*     immt2@    imtmx@     imwtmc@   sact@      sua1@
cphead*    imextract* immua1@    imty2@     imwtmx@   scat*      sua2@
crlf*      imfill*    immua2@    imua1@     imwty2@   sethead*   sub1@
delhead*   imgsc@     immub1@    imua2@     imwua1@   setpix*    subpix*

```

```

delwcs*      imgsc2@      immucac1@    imub1@      imwua2@     sgsc@       sucac1@
edhead*     imgsc@       immucac2@    imucac1@    imwub1@     sgsc2@      sucac2@
filename*   imhead*     immujc@      imucac2@    imwucac1@   sgsc@       sujc@
fileroot*   imiras@     immusa1@     imujc@      imwucac2@   ship@       sumpix*
filext*     immact@     immusa2@     imusa1@     imwujc@     simpos*     susa1@
fixpix*     immatch*    imppm@       imusa2@     imwusa1@    siras@      susa2@
getcol*     immgsc@     imresize*    imwact@     imwusa2@    sky2xy*     wcshead*
getdate*    immgsc2@    imrot*       imwcs*      isfile*     skycoor*    xy2sky*
getfits*    immgsc@     imsao@       imwgsc@     isfits*     sp2char*
gethead*    immiras@    imsdss@      imwgsc2@    isnum*      sppm@
getpix*     immppm@     imsize*      imwgsca@    isrange*    ssao@
# exit
exit
% cd ..

```

3 Trying WCSTools

3.1 Downloading PPM Catalog

Download PPM Catalog.

```

% curl -o ppm.tar.gz http://tdc-www.harvard.edu/software/catalogs/ppm.tar.gz
% Total      % Received % Xferd Average Speed   Time    Time     Time  Current
             Dload  Upload   Total   Spent    Left  Speed
100 19.1M  100 19.1M    0     0  608k      0  0:00:32  0:00:32 --:--:--  730k
% ls -l ppm.tar.gz
-rw-r--r--  1 daisuke taiwan  20127548 Jun  8 12:45 ppm.tar.gz

```

3.2 Installation of PPM Catalog

Install PPM Catalog.

```

% su
Password:
# tar xzvf ppm.tar.gz
x PPM
x PPMra
# ls -l PPM*
-rw-r--r--  1 114  _rtadvd  10609508 Jul 28  1993 PPM
-rw-r--r--  1 114  _rtadvd  12125148 Jul 28  1993 PPMra
# chown root:wheel PPM*
# ls -l PPM*
-rw-r--r--  1 root  wheel  10609508 Jul 28  1993 PPM
-rw-r--r--  1 root  wheel  12125148 Jul 28  1993 PPMra
# mkdir -p /usr/local/share/astro/catalog/ppm
# mv -i PPM* /usr/local/share/astro/catalog/ppm/
# exit
exit
% ls
ppm.tar.gz          wcstools-3.9.6/          wcstools-3.9.6.tar.gz

```

3.3 Setting up PPM Catalog

Set an environment variable for the use of PPM Catalog. If you are using `csh` or its variants, then try following.

```
% setenv PPM_PATH /usr/local/share/astro/catalog/ppm
```

If you are using `sh` or its variants, then try following.

```
% export PPM_PATH=/usr/local/share/astro/catalog/ppm
```

3.4 Trying `scat` command

Use the command `scat` and show a list of stars around Vega.

```
% scat -c ppm -r 1800 18:36:56 38:47:01
081558 18:36:56.332 +38:47:01.17 0.00 A0 3.89
081526 18:35:21.480 +38:53:41.73 6.90 A0 1174.79
081554 18:36:48.698 +38:31:15.07 8.00 F5 949.79
081529 18:35:36.774 +39:03:42.33 9.30 G0 1362.89
081523 18:35:17.499 +39:03:17.04 10.40 K0 1508.02
081579 18:38:00.570 +38:45:48.38 10.40 K 758.59
081602 18:38:59.344 +39:01:31.66 10.50 A5 1682.56
081557 18:36:56.128 +38:45:44.83 10.60 -- 76.18
081552 18:36:40.625 +38:41:29.29 10.70 -- 377.35
081549 18:36:31.914 +38:30:15.98 11.00 K 1043.88
```

4 Preparing your own catalogue for `WCSTools`

4.1 Downloading Gaia Early-DR3

Make a Python script to download a tiny portion of GAIA Early-DR3.

Python Code 1: `ao2021_s22_01.py`

```
#!/usr/pkg/bin/python3.9

# importing astropy module
import astropy.coordinates
import astropy.units

# importing astroquery module
import astroquery.gaia

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# units
u_ha = astropy.units.hourangle
u_deg = astropy.units.deg

# coordinate
coord_ra = "18:36:56.3"
coord_dec = "+38:47:01"

# search radius in deg
radius_deg = 0.1
```

```

# output file name
file_output = 'sample_gaia_edr3.vot.gz'

# making skycoord object
coord = astropy.coordinates.SkyCoord (coord_ra, coord_dec, \
                                       unit=(u_ha, u_deg), frame='icrs')

# RA and Dec in deg
coord_ra_deg = coord.ra.deg
coord_dec_deg = coord.dec.deg

# command for database query
query_p = "POINT('ICRS',gaiaedr3.gaia_source.ra,gaiaedr3.gaia_source.dec)"
query_c = "CIRCLE('ICRS',%f,%f,%f)" % (coord_ra_deg, coord_dec_deg, radius_deg)
query   = "SELECT * from gaiaedr3.gaia_source WHERE CONTAINS(%s,%s)=1;" \
         % (query_p, query_c)

# sending a job to Gaia database
job = astroquery.gaia.Gaia.launch_job_async (query, dump_to_file=True, \
                                             output_file=file_output, \
                                             output_format='votable')

print (job)

# getting results
results = job.get_results ()

# printing results
print (results)

```

Execute the script, and download a part of Gaia EDR3 data.

```

% chmod a+x ao2021_s22_01.py
% ./ao2021_s22_01.py
Created TAP+ (v20200428.1) - Connection:
  Host: gea.esac.esa.int
  Use HTTPS: True
  Port: 443
  SSL Port: 443
Created TAP+ (v20200428.1) - Connection:
  Host: gea.esac.esa.int
  Use HTTPS: True
  Port: 443
  SSL Port: 443
Saving results to: sample_gaia_edr3.vot.gz
Jobid: 16231445918390
Phase: COMPLETED
Owner: None
Output file: sample_gaia_edr3.vot.gz
Results: None

```

solution_id	designation	...	ecl_lat
		...	deg
1636042515805110273	Gaia EDR3 2097890966305682176	...	61.6396324880935
1636042515805110273	Gaia EDR3 2097890966305682432	...	61.64219341638623
1636042515805110273	Gaia EDR3 2097890970599942016	...	61.63834882665206
1636042515805110273	Gaia EDR3 2097890970599947776	...	61.640040171504424
1636042515805110273	Gaia EDR3 2097891039319356160	...	61.63354630720647


```

1636042515805110273 Gaia EDR3 2097891039323180160 ... 61.634598447332856
...
1636042515805110273 Gaia EDR3 2097942063532489216 ... 61.82597603444432
1636042515805110273 Gaia EDR3 2097942063532491520 ... 61.82650177412783
1636042515805110273 Gaia EDR3 2097942063532616960 ... 61.82750154569717
1636042515805110273 Gaia EDR3 2097942063535022592 ... 61.81697718308908
1636042515805110273 Gaia EDR3 2097942127955429888 ... 61.82250743116177
1636042515805110273 Gaia EDR3 2097942127956129536 ... 61.8180196168135
1636042515805110273 Gaia EDR3 2097942166611712000 ... 61.82155296328496
Length = 739 rows
% ls -l sample_gaia_edr3.vot.gz
-rw-r--r-- 1 daisuke taiwan 272010 Jun 8 17:29 sample_gaia_edr3.vot.gz

```

4.2 Reading VOTable file

Make a Python script to read a VOTable file.

Python Code 2: ao2021_s22_02.py

```

#!/usr/pkg/bin/python3.9

# importing argparse module
import argparse

# importing sys module
import sys

# importing numpy module
import numpy

# importing astropy module
import astropy.io.votable

# command-line argument analysis
parser = argparse.ArgumentParser (description='reading VOTable file')
parser.add_argument ('-i', '--input', help='input VOTable file name')
args = parser.parse_args ()

# VOTable file name
file_votable = args.input

# check of VOTable file name
if not ( (file_votable[-4:] == '.vot') or (file_votable[-7:] == '.vot.gz') ):
    print ("Input file must be VOTable file (*.vot or *.vot.gz)")
    sys.exit ()

# reading VOTable
table = astropy.io.votable.parse_single_table (file_votable).to_table ()

# data
data_id      = numpy.array (table['source_id'])
data_desig   = numpy.array (table['designation'])
data_ra      = numpy.array (table['ra'])
data_dec     = numpy.array (table['dec'])
data_parallax = numpy.array (table['parallax'])
data_pmra    = numpy.array (table['pmra'])
data_pmdec   = numpy.array (table['pmdec'])
data_b       = numpy.array (table['phot_bp_mean_mag'])
data_g       = numpy.array (table['phot_g_mean_mag'])

```

```

data_r      = numpy.array (table['phot_rp_mean_mag'])
data_br     = numpy.array (table['bp_rp'])
data_bg     = numpy.array (table['bp_g'])
data_gr     = numpy.array (table['g_rp'])
data_ra_err = numpy.array (table['ra_error'])
data_dec_err = numpy.array (table['dec_error'])
data_pmra_err = numpy.array (table['pmra_error'])
data_pmdec_err = numpy.array (table['pmdec_error'])
data_p_snr  = numpy.array (table['parallax_over_error'])
data_b_snr  = numpy.array (table['phot_bp_mean_flux_over_error'])
data_g_snr  = numpy.array (table['phot_g_mean_flux_over_error'])
data_r_snr  = numpy.array (table['phot_rp_mean_flux_over_error'])

# printing data
i_max = 10
for i in range ( len (data_id) ):
    # stop the loop, if i >= i_max
    if (i >= i_max):
        break
    # printing information
    print ("%s %12.8f %+13.8f %7.3f" \
           % (data_id[i], data_ra[i], data_dec[i], data_r[i]) )

```

Run the script.

```

% chmod a+x ao2021_s22_02.py
% ./ao2021_s22_02.py -h
usage: ao2021_s22_02.py [-h] [-i INPUT]

reading VOTable file

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, --input INPUT
                        input VOTable file name

% ./ao2021_s22_02.py -i sample_gaia_edr3.vot.gz
2097890966305682176 279.28439978 +38.69488367 16.778
2097890966305682432 279.29038600 +38.69810559 16.936
2097890970599942016 279.29095764 +38.69428625 19.513
2097890970599947776 279.29613583 +38.69654488 20.153
2097891039319356160 279.25665798 +38.68579361 19.444
2097891039323180160 279.25467604 +38.68664517 19.223
2097891069384127232 279.24229724 +38.68476898 17.237
2097891073682912000 279.23844362 +38.68472670 18.967
2097891073684549120 279.23767646 +38.68522719 19.747
2097891108038836352 279.25161713 +38.69377980 19.116

```

4.3 Making a local catalog file for WCSTools

Make a Python script to generate a local catalogue file for WCSTools.

Python Code 3: ao2021_s22_03.py

```

#!/usr/pkg/bin/python3.9

# importing argparse module
import argparse

```

```
# importing sys module
import sys

# importing numpy module
import numpy

# importing astropy module
import astropy.io.votable

# command-line argument analysis
parser = argparse.ArgumentParser (description='reading VOTable file')
parser.add_argument ('-i', '--input', help='input VOTable file name')
parser.add_argument ('-o', '--output', help='output catalogue file name')
args = parser.parse_args ()

# VOTable file name
file_votable = args.input
file_catalogue = args.output

# check of VOTable file name
if not ( (file_votable[-4:] == '.vot') or (file_votable[-7:] == '.vot.gz') ):
    print ("Input file must be VOTable file (*.vot or *.vot.gz)")
    sys.exit ()

# check of catalogue file name
if not (file_catalogue[-4:] == '.cat'):
    print ("Output file must be catalogue file (*.cat)")
    sys.exit ()

# reading VOTable
table = astropy.io.votable.parse_single_table (file_votable).to_table ()

# data
data_id = numpy.array (table['source_id'])
data_desig = numpy.array (table['designation'])
data_ra = numpy.array (table['ra'])
data_dec = numpy.array (table['dec'])
data_parallax = numpy.array (table['parallax'])
data_pmra = numpy.array (table['pmra'])
data_pmdec = numpy.array (table['pmdec'])
data_b = numpy.array (table['phot_bp_mean_mag'])
data_g = numpy.array (table['phot_g_mean_mag'])
data_r = numpy.array (table['phot_rp_mean_mag'])
data_br = numpy.array (table['bp_rp'])
data_bg = numpy.array (table['bp_g'])
data_gr = numpy.array (table['g_rp'])
data_ra_err = numpy.array (table['ra_error'])
data_dec_err = numpy.array (table['dec_error'])
data_pmra_err = numpy.array (table['pmra_error'])
data_pmdec_err = numpy.array (table['pmdec_error'])
data_p_snr = numpy.array (table['parallax_over_error'])
data_b_snr = numpy.array (table['phot_bp_mean_flux_over_error'])
data_g_snr = numpy.array (table['phot_g_mean_flux_over_error'])
data_r_snr = numpy.array (table['phot_rp_mean_flux_over_error'])

# writing data to file
with open (file_catalogue, 'w') as fh:
    # printing header
```

```

fh.write ("gaia_edr3/d/j\n")
fh.write ("a tiny portion of Gaia EDR3\n")
# writing each object
for i in range ( len (data_id) ):
    if ( numpy.isnan (data_r[i]) ):
        fh.write ("%012d %12.8f %+13.8f %7.3f\n" \
                % (i, data_ra[i], data_dec[i], 99.999) )
    else:
        fh.write ("%012d %12.8f %+13.8f %7.3f\n" \
                % (i, data_ra[i], data_dec[i], data_r[i]) )

```

Execute the script, and make a local catalogue file.

```

% chmod a+x ao2021_s22_03.py
% ./ao2021_s22_03.py -h
usage: ao2021_s22_03.py [-h] [-i INPUT] [-o OUTPUT]

reading VOTable file

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, --input INPUT
                        input VOTable file name
  -o OUTPUT, --output OUTPUT
                        output catalogue file name

% ./ao2021_s22_03.py -i sample_gaia_edr3.vot.gz -o sample_gaia_edr3.cat
% ls -l sample_gaia_edr3.*
-rw-r--r--  1 daisuke taiwan   35514 Jun  8 18:46 sample_gaia_edr3.cat
-rw-r--r--  1 daisuke taiwan  272010 Jun  8 17:29 sample_gaia_edr3.vot.gz
head sample_gaia_edr3.cat
gaia_edr3/d/j
a tiny portion of Gaia EDR3
000000000000 279.28439978 +38.69488367 16.778
000000000001 279.29038600 +38.69810559 16.936
000000000002 279.29095764 +38.69428625 19.513
000000000003 279.29613583 +38.69654488 20.153
000000000004 279.25665798 +38.68579361 19.444
000000000005 279.25467604 +38.68664517 19.223
000000000006 279.24229724 +38.68476898 17.237
000000000007 279.23844362 +38.68472670 18.967

```

4.4 Trying WCSTools using a local catalogue file

Use the command `scat` to search for stars around Vega.

```

% scat -c sample_gaia_edr3.cat -r 90 18:36:56 38:47:01
000000000137 18:36:56.101 +38:45:43.37 8.95 77.64
000000000441 18:36:50.243 +38:46:44.29 12.89 69.36
000000000443 18:36:51.519 +38:47:10.62 13.80 53.27
000000000295 18:36:58.295 +38:47:20.79 14.35 33.35
000000000565 18:36:53.831 +38:47:21.67 14.45 32.72
000000000136 18:36:55.416 +38:46:29.77 14.96 31.97
000000000566 18:36:53.192 +38:47:37.95 15.01 49.42
000000000439 18:36:53.343 +38:46:29.03 15.31 44.58
000000000294 18:36:59.043 +38:46:26.57 15.41 49.51
000000000437 18:36:49.664 +38:46:35.63 15.57 78.32

```

```

000000000436 18:36:51.795 +38:45:57.36 15.62 80.42
000000000449 18:36:49.793 +38:46:52.84 15.71 73.03
000000000570 18:36:53.746 +38:47:54.09 16.27 59.27
000000000306 18:37:02.160 +38:47:46.42 16.60 85.15
000000000573 18:36:51.142 +38:47:56.43 16.67 79.37
000000000134 18:36:55.425 +38:45:45.73 16.70 75.57
000000000135 18:36:56.109 +38:45:53.97 16.77 67.04
000000000447 18:36:49.377 +38:47:26.52 16.98 81.53
000000000440 18:36:52.163 +38:46:23.73 17.10 58.33
000000000442 18:36:50.936 +38:46:41.63 17.94 62.30
000000000574 18:36:53.260 +38:48:22.12 18.22 87.21
000000000576 18:36:54.762 +38:48:14.35 18.45 74.76
000000000311 18:37:01.850 +38:47:58.30 18.52 89.22
000000000438 18:36:50.731 +38:46:33.61 18.52 67.43
000000000296 18:37:00.517 +38:47:33.00 18.68 61.75
000000000293 18:37:02.958 +38:46:42.77 18.70 83.38
000000000129 18:36:58.247 +38:45:35.13 18.93 89.80
000000000448 18:36:49.746 +38:47:35.52 19.01 80.86
000000000572 18:36:53.926 +38:48:16.59 19.19 79.39
000000000435 18:36:49.315 +38:46:29.10 19.43 84.43
000000000292 18:37:03.565 +38:46:47.27 19.97 89.52
000000000445 18:36:49.789 +38:46:55.06 100.00 72.87

```

5 Preparing data

5.1 Making a symbolic link

Make a symbolic link pointing to the directory that contains reduced data taken on 14/Feb/2021.

```

% ln -s ../../14_reduction3/script_14/ccdred_20210511_122633 0214red
% ls
0214red@          ao2021_s22_02.py~*          sample_gaia_edr3.cat
ao2021_s22_01.py*  ao2021_s22_03.py*          sample_gaia_edr3.vot.gz
ao2021_s22_01.py~  ao2021_s22_03.py~*          wcstools-3.9.6/
ao2021_s22_02.py*  ppm.tar.gz                  wcstools-3.9.6.tar.gz
% ls 0214red/
ccdred.log          lot_20210214_0229_df.fits
dark_00005000.fits  lot_20210214_0230_df.fits
dark_00010000.fits  lot_20210214_0231_df.fits
dark_00015000.fits  lot_20210214_0232_df.fits
dark_00045000.fits  lot_20210214_0233_df.fits
dark_00060000.fits  lot_20210214_0237_df.fits
dark_00180000.fits  lot_20210214_0238_df.fits
flat_gp_Astrodon_2019.fits  lot_20210214_0239_df.fits
flat_ip_Astrodon_2019.fits  lot_20210214_0240_df.fits
flat_rp_Astrodon_2019.fits  lot_20210214_0241_df.fits
.....
lot_20210214_0213_df.fits  lot_20210214_0379_df.fits
lot_20210214_0217_df.fits  lot_20210214_0380_df.fits
lot_20210214_0218_df.fits  lot_20210214_0381_df.fits
lot_20210214_0219_df.fits  lot_20210214_0382_df.fits
lot_20210214_0220_df.fits  lot_20210214_0386_df.fits
lot_20210214_0221_df.fits  lot_20210214_0387_df.fits
lot_20210214_0225_df.fits  lot_20210214_0388_df.fits
lot_20210214_0226_df.fits  lot_20210214_0389_df.fits
lot_20210214_0227_df.fits  lot_20210214_0390_df.fits

```

```
lot_20210214_0228_df.fits
```

5.2 Choosing and copying FITS files

Make a Python script which works like “gethead” command of WCSTools.

Python Code 4: ao2021_s22_04.py

```
#!/usr/pkg/bin/python3.9

# importing argparse module
import argparse

# importing sys module
import sys

# importing pathlib module
import pathlib

# importing astropy module
import astropy.io.fits

# constructing parser object
desc = "a utility like gethead of WCSTools"
parser = argparse.ArgumentParser (description=desc)

# adding arguments
parser.add_argument ('-k', '--keyword', nargs='+', \
                    help='FITS keywords (e.g. "TIME-OBS EXPTIME FILTER)')
parser.add_argument ('files', nargs='+', help='FITS files')

# command-line argument analysis
args = parser.parse_args ()

# input parameters
list_keyword = args.keyword
files_fits   = args.files

# function to read header of a FITS file
def read_fits_header (file_fits):
    # opening FITS file
    with astropy.io.fits.open (file_fits) as hdu:
        # reading header and image
        header = hdu[0].header
        # if no image in PrimaryHDU, then read next HDU
        if (header['NAXIS'] == 0):
            header = hdu[1].header
        # returning header and image
        return (header)

# printing a header
first_line = "# file name"
for keyword in list_keyword:
    first_line = first_line + ", " + keyword
print (first_line)

# processing FITS files one-by-one
for file_fits in files_fits:
    # check of file name
```

```

if not (file_fits[-5:] == '.fits'):
    # printing a message to stderr
    print ("Input file \"%s\" is not a FITS file." % file_fits, \
          file=sys.stderr)
    print ("Input file must be \"*.fits\".")
    # skipping
    continue

# making pathlib object
path_fits = pathlib.Path (file_fits)
filename = path_fits.name

# if the file does not exist, then skip
if not ( path_fits.exists () and path_fits.is_file () ):
    # printing a message to stderr
    print ("The file \"%s\" does not exist or is not a regular file." \
          % file_fits, file=sys.stderr)
    # skipping
    continue

# reading FITS header
header = read_fits_header (file_fits)

# making an empty list for storing values of keywords
list_values = []

# values of keywords
for keyword in list_keyword:
    # if the keyword exists, then read the value from FITS header
    if (keyword in header):
        value = header[keyword]
    # if the keyword does not exists, then use '__NONE__' as a value
    else:
        value = '__NONE__'
    # appending the value to the list
    list_values.append (value)

# printing information
data = str (filename)
for value in list_values:
    data = data + " " + str (value)
print (data)

```

Execute the script.

```

% chmod a+x ao2021_s22_04.py
% ./ao2021_s22_04.py -h
usage: ao2021_s22_04.py [-h] [-k KEYWORD [KEYWORD ...]] files [files ...]

a utility like gethead of WCSTools

positional arguments:
  files                FITS files

optional arguments:
  -h, --help          show this help message and exit
  -k KEYWORD [KEYWORD ...], --keyword KEYWORD [KEYWORD ...]
                    FITS keywords (e.g. "TIME-OBS EXPTIME FILTER")

```

```

% ./ao2021_s22_04.py 0214red/*.fits -k time-obs imagetyp exptime filter object
# file name, time-obs, imagetyp, exptime, filter, object
dark_00005000.fits 22:27:01 DARK 5.0 __NONE__ domeflat
dark_00010000.fits 22:32:02 DARK 10.0 __NONE__ domeflat
dark_00015000.fits 22:38:46 DARK 15.0 __NONE__ domeflat
dark_00045000.fits 23:10:34 DARK 45.0 __NONE__ domeflat
dark_00060000.fits 23:28:55 DARK 60.0 __NONE__ domeflat
dark_00180000.fits 01:09:05 DARK 180.0 __NONE__ domeflat
flat_gp_Astrodon_2019.fits 21:35:45 FLAT 45.0 gp_Astrodon_2019 dark
flat_ip_Astrodon_2019.fits 20:43:23 FLAT 60.0 ip_Astrodon_2019 dark
flat_rp_Astrodon_2019.fits 21:37:34 FLAT 45.0 rp_Astrodon_2019 dark
lot_20210214_0085_df.fits 16:23:10 LIGHT 60.0 rp_Astrodon_2019 V0678VIR

.....

lot_20210214_0378_df.fits 21:06:31 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0379_df.fits 21:07:47 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0380_df.fits 21:09:06 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0381_df.fits 21:10:22 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0382_df.fits 21:11:38 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0386_df.fits 21:14:26 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0387_df.fits 21:15:41 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0388_df.fits 21:16:56 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0389_df.fits 21:18:10 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0390_df.fits 21:19:24 LIGHT 60.0 rp_Astrodon_2019 V0678VIR

```

For this session, we process the data of the variable star V0678 Vir. List FITS files of V0678 Vir data.

```

% ./ao2021_s22_04.py 0214red/*.fits -k time-obs imagetyp exptime filter object \
? | grep V0678VIR
lot_20210214_0085_df.fits 16:23:10 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0086_df.fits 16:24:26 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0087_df.fits 16:25:40 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0088_df.fits 16:26:54 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0089_df.fits 16:28:09 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0117_df.fits 16:53:17 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0118_df.fits 16:54:35 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0119_df.fits 16:55:52 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0120_df.fits 16:57:10 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0121_df.fits 16:58:27 LIGHT 60.0 rp_Astrodon_2019 V0678VIR

.....

lot_20210214_0378_df.fits 21:06:31 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0379_df.fits 21:07:47 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0380_df.fits 21:09:06 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0381_df.fits 21:10:22 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0382_df.fits 21:11:38 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0386_df.fits 21:14:26 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0387_df.fits 21:15:41 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0388_df.fits 21:16:56 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0389_df.fits 21:18:10 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
lot_20210214_0390_df.fits 21:19:24 LIGHT 60.0 rp_Astrodon_2019 V0678VIR
% ./ao2021_s21_01.py 0214red/*.fits -k time-obs imagetyp exptime filter object \
? | grep V0678VIR | wc
      81      486     5832

```


81 files are found.

Check whether or not all the data are acquired using r'-band filter.

```
% ./ao2021_s22_04.py 0214red/*.fits -k time-obs imagety p exptime filter object \
? | grep V0678VIR | grep -v rp_Astrodon_2019
lot_20210214_0337_df.fits 20:25:06 LIGHT 60.0 gp_Astrodon_2019 V0678VIR
lot_20210214_0338_df.fits 20:26:22 LIGHT 60.0 gp_Astrodon_2019 V0678VIR
lot_20210214_0341_df.fits 20:30:20 LIGHT 60.0 ip_Astrodon_2019 V0678VIR
lot_20210214_0342_df.fits 20:31:34 LIGHT 60.0 ip_Astrodon_2019 V0678VIR
```

There are two g'-band data and two more i'-band data.

```
% ./ao2021_s22_04.py 0214red/*.fits -k time-obs imagety p exptime filter object \
? | grep V0678VIR | grep rp_Astrodon_2019 | wc
      77      462     5544
```

Number of r'-band data of V0678 Vir is 77.

5.3 Copying FITS files

Make a Python script to copy FITS files of V0678 Vir data.

Python Code 5: ao2021_s22_05.py

```
#!/usr/pkg/bin/python3.9

# importing sys module
import sys

# importing pathlib module
import pathlib

# importing shutil module
import shutil

# source directory
dir_src = '0214red'

# destination directory
dir_dst = 'v0678vir'

# making a directory if it does not exist
path_dst = pathlib.Path (dir_dst)
path_dst.mkdir (mode=0o755, exist_ok=True)

# reading data from standard input
for line in sys.stdin:
    # if the line starts with '#', then skip
    if (line[0] == '#'):
        continue
    # splitting data
    record = line.split ()
    # data
    filename     = record[0]
    imagetype    = record[1]
    exptime      = float (record[2])
    filtername   = record[3]
```

```

objectname = record[4]
# rejecting data if criteria do not match
if not ( (imagetype == 'LIGHT') and (exptime == 60.0) \
        and (filtername == 'rp_Astrodon_2019') \
        and (objectname == 'V0678VIR') ):
    continue
# copying file
file_fits = "%s/%s" % (dir_src, filename)
path_fits = pathlib.Path (file_fits)
print ("copying file from %s to %s..." % (path_fits, dir_dst) )
shutil.copy2 (path_fits, dir_dst)

```

Run the script and copy FITS files.

```

% chmod a+x ao2021_s22_05.py
% ./ao2021_s22_04.py 0214red/*.fits -k imagetyp exptime filter object | \
? ./ao2021_s22_05.py
copying file from 0214red/lot_20210214_0085_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0086_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0087_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0088_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0089_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0117_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0118_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0119_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0120_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0121_df.fits to v0678vir...

.....

copying file from 0214red/lot_20210214_0378_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0379_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0380_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0381_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0382_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0386_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0387_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0388_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0389_df.fits to v0678vir...
copying file from 0214red/lot_20210214_0390_df.fits to v0678vir...
% ls
0214red@          ao2021_s22_03.py*          sample_gaia_edr3.cat
ao2021_s22_01.py*  ao2021_s22_03.py~*       sample_gaia_edr3.vot.gz
ao2021_s22_01.py~  ao2021_s22_04.py*       v0678vir/
ao2021_s22_02.py*  ao2021_s22_05.py*       wcstools-3.9.6/
ao2021_s22_02.py~* ppm.tar.gz               wcstools-3.9.6.tar.gz
% ls v0678vir/
lot_20210214_0085_df.fits          lot_20210214_0241_df.fits
lot_20210214_0086_df.fits          lot_20210214_0257_df.fits
lot_20210214_0087_df.fits          lot_20210214_0258_df.fits
lot_20210214_0088_df.fits          lot_20210214_0259_df.fits
lot_20210214_0089_df.fits          lot_20210214_0260_df.fits
lot_20210214_0117_df.fits          lot_20210214_0261_df.fits
lot_20210214_0118_df.fits          lot_20210214_0277_df.fits
lot_20210214_0119_df.fits          lot_20210214_0278_df.fits
lot_20210214_0120_df.fits          lot_20210214_0279_df.fits
lot_20210214_0121_df.fits          lot_20210214_0280_df.fits

```

```

.....
lot_20210214_0198_df.fits          lot_20210214_0379_df.fits
lot_20210214_0217_df.fits          lot_20210214_0380_df.fits
lot_20210214_0218_df.fits          lot_20210214_0381_df.fits
lot_20210214_0219_df.fits          lot_20210214_0382_df.fits
lot_20210214_0220_df.fits          lot_20210214_0386_df.fits
lot_20210214_0221_df.fits          lot_20210214_0387_df.fits
lot_20210214_0237_df.fits          lot_20210214_0388_df.fits
lot_20210214_0238_df.fits          lot_20210214_0389_df.fits
lot_20210214_0239_df.fits          lot_20210214_0390_df.fits
lot_20210214_0240_df.fits
% ls v0678vir/*.fits | wc
      77      77    2695

```

6 Reading a FITS file and printing header

Make a Python script to read a FITS file and print its header part.

Python Code 6: ao2021_s22_06.py

```

#!/usr/pkg/bin/python3.9

# importing argparse module
import argparse

# importing sys module
import sys

# importing astropy module
import astropy.io.fits

# command-line argument analysis
desc = 'printing header of a FITS file'
parser = argparse.ArgumentParser (description=desc)
parser.add_argument ('-i', '--file-input', help='input FITS file name')
args = parser.parse_args ()

# FITS file name
file_fits = args.file_input

# check of FITS file name
if not (file_fits[-5:] == '.fits'):
    print ("Input file must be a FITS file")
    sys.exit ()

# function to read header of a FITS file
def read_fits_header (file_fits):
    # opening FITS file
    with astropy.io.fits.open (file_fits) as hdu:
        # reading header and image
        header = hdu[0].header
        # if no image in PrimaryHDU, then read next HDU
        if (header['NAXIS'] == 0):
            header = hdu[1].header
    # returning header and image
    return (header)

```

```
# reading FITS header
header = read_fits_header (file_fits)

# printing FITS header
print (repr (header))
```

Run the script and show the header.

```
% chmod a+x ao2021_s22_06.py
% ./ao2021_s22_06.py -h
usage: ao2021_s22_06.py [-h] [-i FILE_INPUT]

printing header of a FITS file

optional arguments:
  -h, --help            show this help message and exit
  -i FILE_INPUT, --file-input FILE_INPUT
                        input FITS file name

% ./ao2021_s22_06.py -i v0678vir/lot_20210214_0085_df.fits
SIMPLE =                T / conforms to FITS standard
BITPIX =                -64 / array data type
NAXIS =                  2 / number of array dimensions
NAXIS1 =                 2048
NAXIS2 =                 2048
DATE-OBS= '2021-02-14T16:23:10' / [ISO 8601] UTC date/time of exposure start
TIME-OBS= '16:23:10' / [old format] UTC time of exposure start
EXPTIME = 6.000000000000E+001 / [sec] Duration of exposure
EXPOSURE= 6.000000000000E+001 / [sec] Duration of exposure
SET-TEMP= -80.000000000000000 / CCD temperature setpoint in C
CCD-TEMP= -80.000000000000000 / CCD temperature at start of exposure in C
XPIXSZ = 15.000000000000000 / Pixel Width in microns (after binning)
YPIXSZ = 15.000000000000000 / Pixel Height in microns (after binning)
XBINNING= 1 / Binning level along the X-axis
YBINNING= 1 / Binning level along the Y-axis
XORGSUBF= 0 / Subframe X position in binned pixels
YORGSUBF= 0 / Subframe Y position in binned pixels
FILTER = 'rp_Astrodon_2019' / Filter name
IMAGETYP= 'LIGHT' / Type of image
OBJCTRA = '12 31 48.11' / [hms J2000] Target right ascension
OBJCTDEC= '-02 06 02.3' / [dms +N J2000] Target declination
OBJCTALT= 46.3405 / Nominal altitude of center of image
OBJCTAZ = 120.8538 / Nominal azimuth of center of image
OBJCTHA = '-02 25 30.00' / Nominal hour angle of center of image
SITELAT = '23 28 07' / Latitude of the imaging location
SITELONG= '120 52 25' / Longitude of the imaging location
JD = 2459260.1827546298 / Julian Date at start of exposure
JD-HELIO= 2459260.1873254078 / Heliocentric Julian Date at exposure midpoint
AIRMASS = 1.38262415379E+000 / Airmass (multiple of zenithal airmass)
FOCALLEN= 8000.0000000000000 / Focal length of telescope in mm
APTDIA = 1000.0000000000000 / Aperture diameter of telescope in mm
APTAREA = 772124.95592236519 / Aperture area of telescope in mm^2
SWCREATE= 'MaxIm DL Version 5.24 130419 0CYVP' / Name of software that created
the image
OBJECT = 'V0678VIR' / Target object name
TELESCOP= 'LOT' / Telescope name
INSTRUME= 'Driver for Princeton Instruments cameras' / Detector instrument name
OBSERVER= 'lulin' / Observer name
```

```

NOTES      = '          '
DETECTOR= '          '
OWNER      = 'Institute of Astronomy, NCU, Taiwan'
TIMESYS   = 'UTC          ' / Default time system
EQUINX    = '2000        '
EPOCH     = '2000        '
RADECSYS= 'FK5          ' / Equatorial coordinate system
CAMERA    = 'SOPHIA     '
GAIN      = '2          '
SITEELEV=                2862
RMSNOISE= '8.5         '
OPERATOR= 'lulin       '
CTYPE1    = 'RA---TAN' / fastest changing axis name
CTYPE2    = 'DEC--TAN' / slowest changing axis name
FOV       = '13'' 08" x 13'' 08"'
PIXSIZE   = 0.390000000000000001
FLIPSTAT= '          '
SWOWNER   = 'Ming-Hsin Chang' / Licensed owner of software
CSTRETCH= 'Low         ' / Initial display stretch mode
CBLACK    =                725 /Initial display black level in ADUs
CWHITE    =                1208 /Initial display white level in ADUs
PEDESTAL=                0 /Correction to add for zero-based ADU
JD-OBS    =                2459260.1827546
HJD-OBS    =                2459260.1869782
BJD-OBS    =                2459260.1878141
AZIMUTH    =                120.8538
ALTITUDE=                46.3405
HA         = '-02 25 30.00'
HISTORY File was processed by PinPoint 6.1.3 at 2021-02-14T16:24:18
DATE      = '14/02/21' / [old format] UTC date of exposure start
UT        = '16:23:10' / [old format] UTC time of exposure start
ST        = '10 06 18.11' / Local apparent sidereal time of exp. start
LAT-OBS   = 2.346861111111E+001 / [deg +N WGS84] Geodetic latitude
LONG-OBS  = 1.208736111111E+002 / [deg +E WGS84] Geodetic longitude
ALT-OBS   = 2.862000000000E+003 / [metres] Altitude above mean sea level
OBSERVAT= 'NCU Lulin observatory' / Observatory name
RA        = '12 31 48.11' / [hms J2000] Target right ascension
DEC       = '-02 06 02.3' / [dms +N J2000] Target declination
COMMENT Updated on 2021-05-11 12:37:19.347688
COMMENT Image subtraction was carried out
COMMENT Operation: data_ao2021_s13/lot_20210214_0085.fits - ccdred_20210511_1226
COMMENT 33/dark_00060000.fits
COMMENT New file: ccdred_20210511_122633/lot_20210214_0085_d.fits
COMMENT Updated on 2021-05-11 12:37:19.645441
COMMENT Image division was carried out
COMMENT Operation: ccdred_20210511_122633/lot_20210214_0085_d.fits / ccdred_2021
COMMENT 0511_122633/flat_rp_Astrodon_2019.fits
COMMENT New file: ccdred_20210511_122633/lot_20210214_0085_df.fits

```

This FITS file does not have WCS-related keywords in the header. We try to add WCS-related keywords to this FITS file.

7 Checking RA and Dec

Check values of RA and Dec in FITS header.

```

% ./ao2021_s22_04.py v0678vir/lot_20210214_0085_df.fits -k RA DEC
# file name, RA, DEC

```

```
lot_20210214_0085_df.fits 12 31 48.11 -02 06 02.3
```

8 Downloading Gaia EDR3 data

Make a Python script to download Gaia EDR3 around (RA, Dec) = (12:31:48.11, -02:06:02.3).

Python Code 7: ao2021_s22_07.py

```
#!/usr/pkg/bin/python3.9

# importing argparse module
import argparse

# importing astropy module
import astropy.coordinates
import astropy.units

# importing astroquery module
import astroquery.gaia

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# constructing parser object
desc = "Downloading Gaia EDR3"
parser = argparse.ArgumentParser (description=desc)

# adding arguments
parser.add_argument ('-c', '--coordinate', default='', \
                    help='RA and Dec (hh:mm:ss.ss,+/-dd:mm:ss.s)')
parser.add_argument ('-o', '--file-output', default='', \
                    help='output file name (*.cat)')
parser.add_argument ('-r', '--radius', type=float, default=0.1, \
                    help='radius in degree (default: 0.1)')

# command-line argument analysis
args = parser.parse_args ()

# input parameters
coordinate = args.coordinate
file_output = args.file_output
radius_deg = args.radius

# check of coordinate
if not (',' in coordinate):
    print ("Coordinate must be hh:mm:ss.ss,+/-dd:mm:ss.s format.")
    sys.exit ()

# check of output file
if not ( (file_output[-4:] == '.vot') or (file_output[-7:] == '.vot.gz') ):
    print ("Output file must be VOTable file (*.vot or *.vot.gz)")
    sys.exit ()

# units
u_ha = astropy.units.hourangle
```

```

u_deg = astropy.units.deg

# coordinate
(coord_ra, coord_dec) = coordinate.split(',')

# making skycoord object
coord = astropy.coordinates.SkyCoord(coord_ra, coord_dec, \
                                     unit=(u_ha, u_deg), frame='icrs')

# RA and Dec in deg
coord_ra_deg = coord.ra.deg
coord_dec_deg = coord.dec.deg

# command for database query
query_p = "POINT('ICRS',gaiaedr3.gaia_source.ra,gaiaedr3.gaia_source.dec)"
query_c = "CIRCLE('ICRS',%f,%f,%f)" % (coord_ra_deg, coord_dec_deg, radius_deg)
query = "SELECT * from gaiaedr3.gaia_source WHERE CONTAINS(%s,%s)=1;" \
        % (query_p, query_c)

# sending a job to Gaia database
job = astroquery.gaia.Gaia.launch_job_async(query, dump_to_file=True, \
                                           output_file=file_output, \
                                           output_format='votable')

print(job)

# getting results
results = job.get_results()

# printing results
print(results)

```

Execute the script, and download Gaia EDR3 data around the star V0678 Vir.

```

% chmod a+x ao2021_s22_07.py
% ./ao2021_s22_07.py -h
Created TAP+ (v20200428.1) - Connection:
  Host: gea.esac.esa.int
  Use HTTPS: True
  Port: 443
  SSL Port: 443
Created TAP+ (v20200428.1) - Connection:
  Host: gea.esac.esa.int
  Use HTTPS: True
  Port: 443
  SSL Port: 443
usage: ao2021_s22_07.py [-h] [-c COORDINATE] [-o FILE_OUTPUT] [-r RADIUS]

Downloading Gaia EDR3

optional arguments:
  -h, --help            show this help message and exit
  -c COORDINATE, --coordinate COORDINATE
                        RA and Dec (hh:mm:ss.ss,+/-dd:mm:ss.s)
  -o FILE_OUTPUT, --file-output FILE_OUTPUT
                        output file name (*.cat)
  -r RADIUS, --radius RADIUS
                        radius in degree (default: 0.1)

```

```

% ./ao2021_s22_07.py -c 12:31:48.11,-02:06:02.3 -r 0.5 -o v0678vir.vot.gz
Created TAP+ (v20200428.1) - Connection:
  Host: gea.esac.esa.int
  Use HTTPS: True
  Port: 443
  SSL Port: 443
Created TAP+ (v20200428.1) - Connection:
  Host: gea.esac.esa.int
  Use HTTPS: True
  Port: 443
  SSL Port: 443
Saving results to: v0678vir.vot.gz
Jobid: 16231524558270
Phase: COMPLETED
Owner: None
Output file: v0678vir.vot.gz
Results: None
  solution_id          designation          ...          ecl_lat
                                ...          deg
-----
1636042515805110273 Gaia EDR3 3695204666784753024 ... 1.1780991628892545
1636042515805110273 Gaia EDR3 3695204696848942208 ... 1.1970842208695824
1636042515805110273 Gaia EDR3 3695204799934682880 ... 1.208296580299118
1636042515805110273 Gaia EDR3 3695204937366936576 ... 1.1864089778904856
1636042515805110273 Gaia EDR3 3695204937366936832 ... 1.193437726321269
1636042515805110273 Gaia EDR3 3695205006086544256 ... 1.2064069621306117
...
1636042515805110273 Gaia EDR3 3693985029215975552 ... 1.4113403103732618
1636042515805110273 Gaia EDR3 3693985033512196224 ... 1.4140033367358567
1636042515805110273 Gaia EDR3 3693985166660854528 ... 1.4419211760280821
1636042515805110273 Gaia EDR3 3693985205310889344 ... 1.4503708194208988
1636042515805110273 Gaia EDR3 3693987988449695232 ... 1.4076031460119702
1636042515805110273 Gaia EDR3 3693988018513220224 ... 1.429794543354934
1636042515805110273 Gaia EDR3 3693988121592438272 ... 1.4376631192839537
Length = 3189 rows
% ls -l v0678vir.vot.gz
-rw-r--r--  1 daisuke taiwan  1120006 Jun  8 19:41 v0678vir.vot.gz

```

9 Making a local catalogue file for WCSTools

Make a local catalogue file for WCSTools.

```

% ./ao2021_s22_03.py -i v0678vir.vot.gz -o v0678vir.cat
% ls -l v0678vir.*
-rw-r--r--  1 daisuke taiwan  153114 Jun  8 19:42 v0678vir.cat
-rw-r--r--  1 daisuke taiwan  1120006 Jun  8 19:41 v0678vir.vot.gz
% head v0678vir.cat
gaia_edr3/d/j
a tiny portion of Gaia EDR3
000000000000 188.37781403 -2.33298085 14.841
000000000001 188.38966978 -2.31740084 19.039
000000000002 188.40351873 -2.31112870 18.137
000000000003 188.35642287 -2.31478922 18.315
000000000004 188.36654128 -2.31147240 17.433
000000000005 188.38769203 -2.30641334 18.259
000000000006 188.40997200 -2.29130897 13.580
000000000007 188.38831705 -2.29902560 20.296

```


10 Source extraction

Carry out source extraction for the file v0678vir/lot_20210214_0085_df.fits.

Python Code 8: ao2021_s22_08.py

```
#!/usr/pkg/bin/python3.9

# importing argparse module
import argparse

# importing sys module
import sys

# importing datetime module
import datetime

# importing numpy module
import numpy.ma

# importing astropy module
import astropy.convolution
import astropy.io.fits
import astropy.io.votable
import astropy.stats

# importing photutils module
import photutils.segmentation

# date/time
now = datetime.datetime.now ()

# constructing parser object
desc = 'source extraction using image segmentation and deblending'
parser = argparse.ArgumentParser (description=desc)

# adding arguments
parser.add_argument ('-i', '--file-input', default='', \
                    help='input FITS file name')
parser.add_argument ('-o', '--file-output', default='', \
                    help='output VOTable file name')
parser.add_argument ('-t', '--threshold', type=float, default=10.0, \
                    help='detection threshold in sigma (default: 10)')
parser.add_argument ('-u', '--threshold-for-sky', type=float, default=2.0, \
                    help='detection threshold for sky estimate (default: 2)')
parser.add_argument ('-n', '--npixels', type=int, default=5, \
                    help='minimum number of pixels for detection (default: 5)')
parser.add_argument ('-s', '--dilate-size', type=int, default=21, \
                    help='dilate size (default: 21)')
parser.add_argument ('-m', '--maxiters', type=int, default=30, \
                    help='maximum number of iterations (default: 30)')
parser.add_argument ('-r', '--sigma-clipping', type=float, default=4.0, \
                    help='sigma-clipping threshold in sigma (default: 4)')
parser.add_argument ('-k', '--gaussian-fwhm', type=float, default=3.0, \
                    help='Gaussian FWHM in pixel for convolution (default: 3)')
parser.add_argument ('-a', '--kernel-size', type=int, default=3, \
                    help='Gaussian kernel array size in pixel (default: 3)')

# command-line argument analysis
args = parser.parse_args ()
```

```
# file names
file_input = args.file_input
file_output = args.file_output

# input parameters
threshold = args.threshold
threshold_for_sky = args.threshold
npixels = args.npixels
dilate_size = args.dilate_size
maxiters = args.maxiters
rejection = args.sigma_clipping
gaussian_fwhm = args.gaussian_fwhm
kernel_array_size = args.kernel_size

# check of input file name
if not (file_input[-5:] == '.fits'):
    print ("Input file must be a FITS file.")
    sys.exit ()

# check of catalogue file name
if not (file_output[-4:] == '.xym'):
    print ("Output file must be *.xym.")
    sys.exit ()

# opening FITS file
with astropy.io.fits.open (file_input) as hdu:
    # reading header and image
    header = hdu[0].header
    image = hdu[0].data
    # if no image in PrimaryHDU, then read next HDU
    if (header['NAXIS'] == 0):
        header = hdu[1].header
        image = hdu[1].data

# making source mask
source_mask \
    = photutils.segmentation.make_source_mask (image, threshold_for_sky,
                                               npixels=npixels,
                                               sigclip_iters=maxiters,
                                               dilate_size=dilate_size)

# making masked array
image_masked = numpy.ma.array (image, mask=source_mask)

# sigma-clipping
skybg_mean, skybg_median, skybg_stddev \
    = astropy.stats.sigma_clipped_stats (image, sigma=rejection)

# mode calculation using empirical formula
skybg_mode = 3.0 * skybg_median - 2.0 * skybg_mean

# detection threshold in ADU
threshold_adu = skybg_mode + threshold * skybg_stddev

# 2D Gaussian kernel for convolution
gaussian_sigma = gaussian_fwhm * astropy.stats.gaussian_fwhm_to_sigma
kernel = astropy.convolution.Gaussian2DKernel (gaussian_sigma, \
                                               x_size=kernel_array_size, \
```

```

                                                    y_size=kernel_array_size)
kernel.normalize ()

# source detection
image_seg = photutils.segmentation.detect_sources (image, threshold_adu, \
                                                    npixels=npixels, \
                                                    filter_kernel=kernel)

# deblending
image_deblend = photutils.segmentation.deblend_sources (image, image_seg, \
                                                        npixels=npixels, \
                                                        filter_kernel=kernel, \
                                                        nlevels=32, \
                                                        contrast=0.001)

# making a source catalogue
catalogue = photutils.segmentation.SourceCatalog (image, image_deblend)

# making a table
table_sources = catalogue.to_table ()

# writing data to file
with open (file_output, 'w') as fh:
    # for each object
    for i in range ( len (table_sources) ):
        # writing X, Y, and instrumental magnitude
        fh.write ("%10.3f %10.3f %+7.3f\n" \
                  % (table_sources['xcentroid'][i], \
                     table_sources['ycentroid'][i], \
                     -2.5 * numpy.log10 (table_sources['kron_flux'][i])) ) )

```

Run the script, and carry out source extraction.

```

% chmod a+x ao2021_s22_08.py
% ./ao2021_s22_08.py -h
usage: ao2021_s22_08.py [-h] [-i FILE_INPUT] [-o FILE_OUTPUT] [-t THRESHOLD]
                        [-u THRESHOLD_FOR_SKY] [-n NPIXELS] [-s DILATE_SIZE]
                        [-m MAXITERS] [-r SIGMA_CLIPPING] [-k GAUSSIAN_FWHM]
                        [-a KERNEL_SIZE]

source extraction using image segmentation and deblending

optional arguments:
  -h, --help            show this help message and exit
  -i FILE_INPUT, --file-input FILE_INPUT
                        input FITS file name
  -o FILE_OUTPUT, --file-output FILE_OUTPUT
                        output VOTable file name
  -t THRESHOLD, --threshold THRESHOLD
                        detection threshold in sigma (default: 10)
  -u THRESHOLD_FOR_SKY, --threshold-for-sky THRESHOLD_FOR_SKY
                        detection threshold for sky estimate (default: 2)
  -n NPIXELS, --npixels NPIXELS
                        minimum number of pixels for detection (default: 5)
  -s DILATE_SIZE, --dilate-size DILATE_SIZE
                        dilate size (default: 21)
  -m MAXITERS, --maxiters MAXITERS
                        maximum number of iterations (default: 30)

```

```

-r SIGMA_CLIPPING, --sigma-clipping SIGMA_CLIPPING
    sigma-clipping threshold in sigma (default: 4)
-k GAUSSIAN_FWHM, --gaussian-fwhm GAUSSIAN_FWHM
    Gaussian FWHM in pixel for convolution (default: 3)
-a KERNEL_SIZE, --kernel-size KERNEL_SIZE
    Gaussian kernel array size in pixel (default: 3)

% ./ao2021_s22_08.py -i v0678vir/lot_20210214_0085_df.fits -o 0085.xym
% ls -l 0085.xym
-rw-r--r--  1 daisuke  taiwan  1860 Jun  8 20:00 0085.xym
% head 0085.xym
1211.632      20.468 -11.545
1355.428      26.018 -13.490
1492.723      29.042 -12.070
1862.156      78.991 -13.645
1407.155      78.823  -9.969
1359.965      88.313 -13.675
 265.389     152.804 -10.402
  12.951     187.371 -13.555
1053.991     187.713 -10.237
 895.333     200.771 -11.856

```

11 Modifying some FITS keywords

Make a Python script to modify some FITS keywords for WCSTools.

Python Code 9: ao2021_s22_09.py

```

#!/usr/pkg/bin/python3.9

# importing argparse module
import argparse

# importing sys module
import sys

# importing datetime module
import datetime

# importing numpy
import numpy

# importing astropy module
import astropy.coordinates
import astropy.io.fits
import astropy.units

# date/time
now = datetime.datetime.now ()

# constructing parser object
desc = 'adding FITS keywords EPOCH and SECPIX to a FITS file'
parser = argparse.ArgumentParser (description=desc)

# adding arguments
parser.add_argument ('-i', '--file-input', default='', \
                    help='input FITS file name')
parser.add_argument ('-o', '--file-output', default='', \

```

```

        help='output FITS file name')
parser.add_argument ('-e1', '--keyword-epoch', default='EPOCH', \
                    help='FITS keyword for epoch (default: EPOCH)')
parser.add_argument ('-e2', '--keyword-equinox', default='EQUINOX', \
                    help='FITS keyword for equinox (default: EQUINOX)')
parser.add_argument ('-f', '--keyword-focallength', default='FOCALLEN', \
                    help='FITS keyword for focal length (default: FOCALLEN)')
parser.add_argument ('-s1', '--keyword-pixelscale1', default='SECPPIX1', \
                    help='FITS keyword for pixel scale (default: SECPPIX1)')
parser.add_argument ('-s2', '--keyword-pixelscale2', default='SECPPIX2', \
                    help='FITS keyword for pixel scale (default: SECPPIX2)')
parser.add_argument ('-x', '--keyword-ypixsz', default='XPIXSZ', \
                    help='FITS keyword for pixel size in X (default: XPIXSZ)')
parser.add_argument ('-y', '--keyword-ypixsz', default='YPIXSZ', \
                    help='FITS keyword for pixel size in Y (default: YPIXSZ)')

# command-line argument analysis
args = parser.parse_args ()

# file names
file_input = args.file_input
file_output = args.file_output

# FITS keywords
keyword_epoch = args.keyword_epoch
keyword_equinox = args.keyword_equinox
keyword_focallength = args.keyword_focallength
keyword_secpix1 = args.keyword_pixelscale1
keyword_secpix2 = args.keyword_pixelscale2
keyword_xpixsz = args.keyword_xpixsz
keyword_ypixsz = args.keyword_ypixsz

# check of input file name
if not (file_input[-5:] == '.fits'):
    print ("Input file must be a FITS file.")
    sys.exit ()

# check of catalogue file name
if not (file_output[-5:] == '.fits'):
    print ("Output file must be a FITS file.")
    sys.exit ()

# units
u_ha = astropy.units.hourangle
u_deg = astropy.units.deg

# function to read a FITS file
def read_fits (file_fits):
    # reading FITS file
    with astropy.io.fits.open (file_fits) as hdu:
        # reading header and image
        header = hdu[0].header
        image = hdu[0].data
        # if no image in PrimaryHDU, then read next HDU
        if (header['NAXIS'] == 0):
            header = hdu[1].header
            image = hdu[1].data
    # returning header and image
    return (header, image)

```

```

# reading header and image of a FITS file
(header, image) = read_fits (file_input)

# focal length in mm
focallength_mm = header[keyword_focallength]
focallength_m = focallength_mm * 10**-3

# pixel sizes
pixelsize_x_micron = header[keyword_xpixsz]
pixelsize_y_micron = header[keyword_ypixsz]
pixelsize_x_m = pixelsize_x_micron * 10**-6
pixelsize_y_m = pixelsize_y_micron * 10**-6

# pixel scale
pixelscale_x = pixelsize_x_m / focallength_m / numpy.pi * 180.0 * 3600.0 * -1.0
pixelscale_y = pixelsize_y_m / focallength_m / numpy.pi * 180.0 * 3600.0

# CDELTA1 and CDELTA2
cdelt1 = pixelsize_x_m / focallength_m / numpy.pi * 180.0 * -1.0
cdelt2 = pixelsize_y_m / focallength_m / numpy.pi * 180.0 * -1.0

# modifying format of RA and DEC value
ra_str = header['RA']
dec_str = header['DEC']
ra_str = ra_str.replace(' ', ':')
dec_str = dec_str.replace(' ', ':')

# RA and Dec
coord = astropy.coordinates.SkyCoord (ra_str, dec_str, unit=(u_ha, u_deg), \
                                     frame='icrs')

ra_deg = coord.ra.deg
dec_deg = coord.dec.deg

# modifying format of HA and ST
ha_str = header['HA']
st_str = header['ST']
ha_str = ha_str.replace(' ', ':')
st_str = st_str.replace(' ', ':')

# updating FITS header
header['comment'] = "Updated on %s" % (now)
header['comment'] = "Added some keywords necessary for WCSTools"
header['comment'] = "New file: %s" % (file_output)
header[keyword_epoch] = 2000.0
header[keyword_equinox] = 2000.0
header[keyword_secpix1] = pixelscale_x
header[keyword_secpix2] = pixelscale_y
header['RA'] = ra_str
header['DEC'] = dec_str
header['HA'] = ha_str
header['ST'] = st_str
header['CRVAL1'] = ra_deg
header['CRVAL2'] = dec_deg
header['CRPIX1'] = header['NAXIS1'] / 2.0
header['CRPIX2'] = header['NAXIS2'] / 2.0
header['CROTA1'] = 0.0
header['CROTA2'] = 0.0
header['CDELTA1'] = cdelt1

```

```
header['CDELTA2']          = cdelt2

# writing a FITS file
astropy.io.fits.writeto (file_output, image, header=header)
```

Execute the script.

```
% chmod a+x ao2021_s22_09.py
% ./ao2021_s22_09.py -h
usage: ao2021_s22_09.py [-h] [-i FILE_INPUT] [-o FILE_OUTPUT]
                        [-e1 KEYWORD_EPOCH] [-e2 KEYWORD_EQUINOX]
                        [-f KEYWORD_FOCALLENGTH] [-s1 KEYWORD_PIXELSCALE1]
                        [-s2 KEYWORD_PIXELSCALE2] [-x KEYWORD_XPIXSZ]
                        [-y KEYWORD_YPIXSZ]

adding FITS keywords EPOCH and SECPIX to a FITS file

optional arguments:
  -h, --help            show this help message and exit
  -i FILE_INPUT, --file-input FILE_INPUT
                        input FITS file name
  -o FILE_OUTPUT, --file-output FILE_OUTPUT
                        output FITS file name
  -e1 KEYWORD_EPOCH, --keyword-epoch KEYWORD_EPOCH
                        FITS keyword for epoch (default: EPOCH)
  -e2 KEYWORD_EQUINOX, --keyword-equinox KEYWORD_EQUINOX
                        FITS keyword for equinox (default: EQUINOX)
  -f KEYWORD_FOCALLENGTH, --keyword-focallength KEYWORD_FOCALLENGTH
                        FITS keyword for focal length (default: FOCALLEN)
  -s1 KEYWORD_PIXELSCALE1, --keyword-pixelscale1 KEYWORD_PIXELSCALE1
                        FITS keyword for pixel scale (default: SECPIX1)
  -s2 KEYWORD_PIXELSCALE2, --keyword-pixelscale2 KEYWORD_PIXELSCALE2
                        FITS keyword for pixel scale (default: SECPIX2)
  -x KEYWORD_XPIXSZ, --keyword-xpixsz KEYWORD_XPIXSZ
                        FITS keyword for pixel size in X (default: XPIXSZ)
  -y KEYWORD_YPIXSZ, --keyword-ypixsz KEYWORD_YPIXSZ
                        FITS keyword for pixel size in Y (default: YPIXSZ)

% ./ao2021_s22_09.py -i v0678vir/lot_20210214_0085_df.fits -o 0085_dfk.fits
% ls -l 0085_dfk.fits
-rw-r--r--  1 daisuke  taiwan  33563520 Jun  8 22:14 0085_dfk.fits
```

Check the header of newly created FITS file.

```
% ./ao2021_s22_06.py -i 0085_dfk.fits
SIMPLE  =                               T / conforms to FITS standard
BITPIX  =                               -64 / array data type
NAXIS   =                               2 / number of array dimensions
NAXIS1  =                               2048
NAXIS2  =                               2048
DATE-OBS= '2021-02-14T16:23:10' / [ISO 8601] UTC date/time of exposure start
TIME-OBS= '16:23:10' / [old format] UTC time of exposure start
EXPTIME = 6.000000000000E+001 / [sec] Duration of exposure
EXPOSURE= 6.000000000000E+001 / [sec] Duration of exposure
SET-TEMP= -80.00000000000000 / CCD temperature setpoint in C

.....
```

```

NOTES      = '          '
DETECTOR= '          '
OWNER      = 'Institute of Astronomy, NCU, Taiwan'
TIMESYS   = 'UTC      ' / Default time system
EQUINX    = '2000     '
EPOCH     =                2000.0
RADECSYS= 'FK5      ' / Equatorial coordinate system
CAMERA    = 'SOPHIA  '
GAIN      = '2       '
SITELEV   =                2862
RMSNOISE= '8.5      '

.....

RA        = '12:31:48.11' / [hms J2000] Target right ascension
DEC       = '-02:06:02.3' / [dms +N J2000] Target declination
EQUINOX   =                2000.0
SECPIX1   = -0.3867465117133057
SECPIX2   =  0.3867465117133057
CRVAL1    =  187.9504583333333
CRVAL2    = -2.100638888888889
CRPIX1    =                1024.0
CRPIX2    =                1024.0
CROTA1    =                0.0
CROTA2    =                0.0
CDELTA1   = -0.00010742958658702
CDELTA2   = -0.00010742958658702
COMMENT   Updated on 2021-05-11 12:37:19.347688
COMMENT   Image subtraction was carried out
COMMENT   Operation: data_ao2021_s13/lot_20210214_0085.fits - ccdred_20210511_1226
COMMENT   33/dark_00060000.fits
COMMENT   New file: ccdred_20210511_122633/lot_20210214_0085_d.fits
COMMENT   Updated on 2021-05-11 12:37:19.645441
COMMENT   Image division was carried out
COMMENT   Operation: ccdred_20210511_122633/lot_20210214_0085_d.fits / ccdred_2021
COMMENT   0511_122633/flat_rp_Astrodon_2019.fits
COMMENT   New file: ccdred_20210511_122633/lot_20210214_0085_df.fits
COMMENT   Updated on 2021-06-08 22:14:45.075041
COMMENT   Added some keywords necessary for WCSTools
COMMENT   New file: 0085_dfk.fits

```

12 Setting WCS of a FITS file

Use the command `imwcs` to set WCS (World Coordinate System) of a FITS file.

```

% imwcs -v -o 0085_dfk.fits -t 5 -d 0085.xym -c v0678vir.cat -h 100 \
? 0085_dfk.fits
0085_dfk.fits:
imwcs IMWCS WCSTools 3.9.6, 31 August 2020, Jessica Mink (jmink@cfa.harvard.edu)
Set World Coordinate System in FITS image file 0085_dfk.fits
CTYPE1 = RA---TAN
CRVAL1 = 187.95045833
CRPIX1 = 1024.00000000
CTYPE2 = DEC--TAN
CRVAL2 = -2.10063889
CRPIX2 = 1024.00000000
CDELTA1 = -0.00010743

```



```

CROTA1 = 0.000
CDELTA2 = -0.00010743
CROTA2 = 0.000
Reference pixel (1024.00,1024.00) 12:31:48.123 -02:06:02.30 J2000
Search at 12:31:48.097 -02:06:02.49 J2000 +- 00:00:26.421 +00:06:36.04
Image width=2048 height=2048, 0.386747 arcsec/pixel
SearchLim: RA: 12:31:21.676 - 12:32:14.519 Dec: -02:12:38.53 - -01:59:26.45
CTGREAD: Catalog v0678vir.cat : 193 / 3190 / 3189 found.cat
CTGREAD: 193 stars found; only 100 returned
Using 100 / 193 reference stars brighter than 18.8
v0678vir.cat:
2475 12:31:42.144 -02:06:25.39 18.84 1255.2 1083.7
DAOREAD: Catalog 0085.xym : 62 / 63 / 62 foundlog 0085.xym
Using brightest 62 / 193 reference stars
Using all 62 image stars
  62 12:31:49.788 -02:03:08.75 -9.93 959.0 575.3 9358921
Match history: nim=62 nref=62 tol= 5 minbin=2 minmatch=31):
.....

2524 12:32:00.636 -02:01:27.01 17.82 442.36 466.42 -11.67 0.06 0.01 0.06
2581 12:31:22.373 -02:03:12.35 17.72 1940.32 713.88 -11.52 0.28 0.38 0.47
2460 12:31:22.168 -02:09:55.10 17.19 1968.76 1764.49 -11.51 0.73 1.64 1.80
2472 12:31:30.558 -02:07:04.67 17.80 1630.87 1324.15 -11.24 -0.23 0.44 0.50
2513 12:32:09.151 -02:03:18.37 18.45 114.85 761.85 -10.95 -0.25 -0.13 0.28
# Mean dx= 0.2354/0.9751 dy= -0.6152/1.4834 dxy= 1.3561
# Mean dra= -0.0948/0.3728 ddec= 0.2348/0.5716 sep= 0.5213/0.6824
# nmatch= 47 nstars= 62 between v0678vir.cat and 0085.xym niter= 1
CTYPE1 = RA---TAN
CRVAL1 = 187.94161508
CRPIX1 = 1024.00000000
CTYPE2 = DEC--TAN
CRVAL2 = -2.08484740
CRPIX2 = 1024.00000000
CD1_1 = -0.00010673215
CD1_2 = 1.966704e-06
CD2_1 = -1.966029e-06
CD2_2 = -0.00010676876
0085_dfkw.fits: written successfully.
% ls -l 0085_dfk*.fits
-rw-r--r-- 1 daisuke taiwan 33563520 Jun 8 22:14 0085_dfk.fits
-rw-r--r-- 1 daisuke taiwan 33566400 Jun 8 22:23 0085_dfkw.fits

```

Refine WCS.

```

% imwcs -v -o -t 3 -d 0085.xym -c v0678vir.cat -h 100 0085_dfkw.fits
0085_dfkw.fits:
imwcs IMWCS WCSTools 3.9.6, 31 August 2020, Jessica Mink (jmink@cfa.harvard.edu)
Set World Coordinate System in FITS image file 0085_dfkw.fits
IMWCS = IMWCS WCSTools 3.9.6, 31 August 2020, Jessica Mink (jmink@cfa.harva
CTYPE1 = RA---TAN
CRVAL1 = 187.94160147
CRPIX1 = 1024.00000000
CTYPE2 = DEC--TAN
CRVAL2 = -2.08479898
CRPIX2 = 1024.00000000
CD1_1 = -0.0001066112
CD1_2 = 1.853013e-06

```

```

CD2_1 = -1.852883e-06
CD2_2 = -0.00010661865
Reference pixel (1024.00,1024.00) 12:31:45.997 -02:05:05.27 J2000
Search at 12:31:45.972 -02:05:05.47 J2000 +- 00:00:26.676 +00:06:39.88
Image width=2048 height=2048, 0.383858 arcsec/pixel
SearchLim: RA: 12:31:19.296 - 12:32:12.647 Dec: -02:11:45.35 - -01:58:25.59
CTGREAD: Catalog v0678vir.cat : 201 / 3190 / 3189 found.cat
CTGREAD: 201 stars found; only 100 returned
Using 100 / 201 reference stars brighter than 18.7
v0678vir.cat:
2630 12:31:26.977 -02:00:55.33 18.73 1754.9 360.1
DAOREAD: Catalog 0085.xym : 62 / 63 / 62 foundlog 0085.xym
Using brightest 62 / 201 reference stars
Using all 62 image stars
  62 12:31:47.449 -02:02:12.61 -9.93 959.0 575.3 9358921
Match history: nim=62 nref=62 tol= 3 minbin=2 minmatch=31):
.....

2640 12:31:40.740 -01:58:41.22 17.43 1211.63 20.47 -11.54 0.10 0.11 0.15
2581 12:31:22.373 -02:03:12.35 17.72 1940.32 713.88 -11.52 -0.17 -0.02 0.17
2460 12:31:22.168 -02:09:55.10 17.19 1968.76 1764.49 -11.51 0.67 0.65 0.94
2472 12:31:30.558 -02:07:04.67 17.80 1630.87 1324.15 -11.24 -0.32 -0.17 0.37
2513 12:32:09.151 -02:03:18.37 18.45 114.85 761.85 -10.95 0.06 0.16 0.17
# Mean dx= -0.0438/0.6925 dy= -0.0118/0.7906 dxy= 0.8793
# Mean dra= 0.0167/0.2667 ddec= 0.0048/0.3027 sep= 0.3375/0.4034
# nmatch= 53 nstars= 62 between v0678vir.cat and 0085.xym niter= 1
IMWCS = IMWCS WCSTools 3.9.6, 31 August 2020, Jessica Mink (jmink@cfa.harva
CTYPE1 = RA---TAN
CRVAL1 = 187.94160358
CRPIX1 = 1024.00000000
CTYPE2 = DEC--TAN
CRVAL2 = -2.08478830
CRPIX2 = 1024.00000000
CD1_1 = -0.00010661868
CD1_2 = 1.858488e-06
CD2_1 = -1.858565e-06
CD2_2 = -0.00010661426
0085_dfkw.fits: rewritten successfully.
% imwcs -v -o -t 1 -d 0085.xym -c v0678vir.cat -h 100 0085_dfkw.fits
0085_dfkw.fits:
imwcs IMWCS WCSTools 3.9.6, 31 August 2020, Jessica Mink (jmink@cfa.harvard.edu)
Set World Coordinate System in FITS image file 0085_dfkw.fits
IMWCS = IMWCS WCSTools 3.9.6, 31 August 2020, Jessica Mink (jmink@cfa.harva
CTYPE1 = RA---TAN
CRVAL1 = 187.94160224
CRPIX1 = 1024.00000000
CTYPE2 = DEC--TAN
CRVAL2 = -2.08479053
CRPIX2 = 1024.00000000
CD1_1 = -0.00010664671
CD1_2 = 1.856987e-06
CD2_1 = -1.856821e-06
CD2_2 = -0.00010665624
Reference pixel (1024.00,1024.00) 12:31:45.997 -02:05:05.24 J2000
Search at 12:31:45.972 -02:05:05.44 J2000 +- 00:00:26.685 +00:06:40.03
Image width=2048 height=2048, 0.383986 arcsec/pixel
SearchLim: RA: 12:31:19.287 - 12:32:12.657 Dec: -02:11:45.48 - -01:58:25.41
CTGREAD: Catalog v0678vir.cat : 201 / 3190 / 3189 found.cat

```

```

CTGREAD: 201 stars found; only 100 returned
Using 100 / 201 reference stars brighter than 18.7
v0678vir.cat:
2630 12:31:26.977 -02:00:55.33 18.73 1754.6 360.4
DAOREAD: Catalog 0085.xym : 62 / 63 / 62 foundlog 0085.xym
Using brightest 62 / 201 reference stars
Using all 62 image stars
 62 12:31:47.450 -02:02:12.51 -9.93 959.0 575.3 9358921
Match history: nim=62 nref=62 tol= 1 minbin=2 minmatch=31):
.....
2524 12:32:00.636 -02:01:27.01 17.82 442.36 466.42 -11.67 0.01 0.24 0.24
2640 12:31:40.740 -01:58:41.22 17.43 1211.63 20.47 -11.54 0.16 -0.06 0.17
2581 12:31:22.373 -02:03:12.35 17.72 1940.32 713.88 -11.52 0.05 -0.06 0.08
2472 12:31:30.558 -02:07:04.67 17.80 1630.87 1324.15 -11.24 -0.17 -0.11 0.20
2513 12:32:09.151 -02:03:18.37 18.45 114.85 761.85 -10.95 -0.12 0.12 0.17
# Mean dx= 0.0173/0.4317 dy= 0.0180/0.5108 dxy= 0.6167
# Mean dra= -0.0065/0.1657 ddec= -0.0070/0.1963 sep= 0.2369/0.2569
# nmatch= 40 nstars= 62 between v0678vir.cat and 0085.xym niter= 1
IMWCS = IMWCS WCSTools 3.9.6, 31 August 2020, Jessica Mink (jmink@cfa.harva
CTYPE1 = RA---TAN
CRVAL1 = 187.94159831
CRPIX1 = 1024.00000000
CTYPE2 = DEC--TAN
CRVAL2 = -2.08479152
CRPIX2 = 1024.00000000
CD1_1 = -0.00010667949
CD1_2 = 1.85958e-06
CD2_1 = -1.859858e-06
CD2_2 = -0.00010666353
0085_dfkw.fits: rewritten successfully.

```

13 Checking the position of V0678 Vir

Use Ginga to check the position of the star V0678 Vir.

```
% ginga 0085_dfkw.fits &
```

Use the function Pick to measure the position of the star V0678 Vir. (Fig. 3) Measured RA and Dec are 12:31:48.070 and -02:06:02.50, respectively.

Download GCVS version 5.1.

```

% curl -o gcvs5.data http://www.sai.msu.su/gcvs/gcvs/gcvs5/gcvs5.txt
% Total      % Received % Xferd  Average Speed   Time    Time     Current
             Dload  Upload   Total   Spent    Left     Speed
100 12.9M  100 12.9M    0     0   383k      0  0:00:34  0:00:34  ---:---:-- 563k
% ls -l gcvs5.data
-rw-r--r--  1 daisuke  taiwan 13608033 Jun  8 22:39 gcvs5.data

```

Search for the star V0678 Vir.

```

% grep 'V0678 Vir' gcvs5.data | cut -b -80
860678 |V0678 Vir |123148.11 -020602.3 |EW          | 14.74  | 15.13  | 15.

```

RA and Dec of V0678 Vir on GCVS 5.1 are 12:31:48.11 and -02:06:02.3, respectively. The accuracy of WCS of 0085_dfkw.fits is good enough for our purpose.

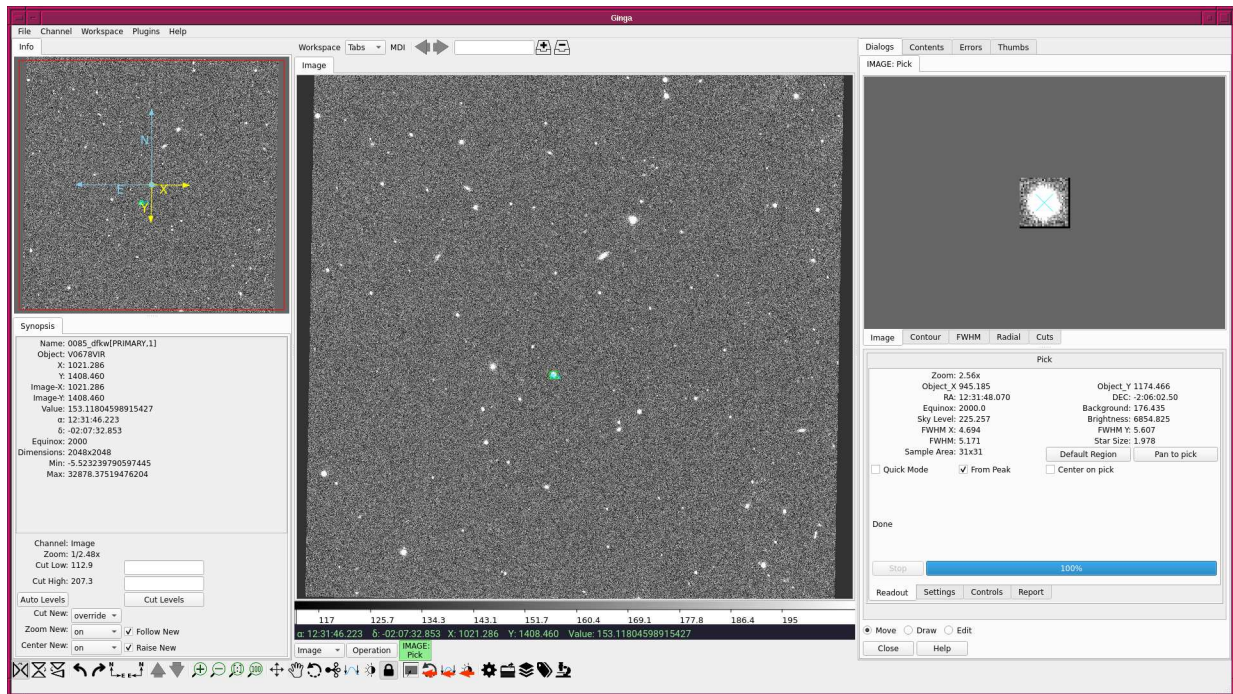


Figure 3: Image of 0085_dfkw.fits.

14 For your training

Read followings.

1. WCSTools

- <http://tdc-www.harvard.edu/software/wcstools/>
- <http://tdc-www.harvard.edu/software/wcstools/publications/adass2018abs.html>
- <http://tdc-www.harvard.edu/software/wcstools/publications/wcstools.adass98.html>

2. FITS World Coordinate System (WCS)

- https://fits.gsfc.nasa.gov/fits_wcs.html

3. FITS World Coordinate Systems

- <https://www.atnf.csiro.au/people/mcalabre/WCS/>

15 Assignment

1. What is Gaia mission?
2. Choose three astrometric catalogues. Describe each catalogue.
3. What is WCS (World Coordinate System)?
4. What are WCS-related FITS keywords? Describe each of them.
5. What is CD matrix?
6. Set WCS for v0678vir/lot_20210214_0085_df.fits using USNO-B1.0 catalogue. Compare the result with the one calibrated by Gaia EDR3.
7. Pick a star on the image 0085_dfkw.fits. Use centroid technique to measure the position of the star. Use the command xy2sky to measure RA and Dec of the star. Describe what you have done.

8. Make your own single Python script which sets WCS of a FITS file. Describe the design of your Python script. Run the script, and show the result. Show source code of your Python script.
9. Set WCS for all the FITS files of r'-band V0678 Vir data taken on 14/Feb/2021.