Checking overlap between Kryukova 2012 + Sokol 2019 and our Mon R2 monitoring

December 19, 2022

[8]:
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import astropy.table
from astropy.coordinates import SkyCoord, Angle
from astropy import units as u

from wuvars.data import spreadsheet, photometry, quality_classes
from wuvars.analysis.luhman16_coord_handler import coords_from_Luhman_table
```

[3]:
```
loc = "/Users/tsrice/Documents/Variability_Project_2020/wuvars/data/
 ↪auxiliary_catalogs/c2d_MonR2/Kryukova_2012_aj428832t1_mrt.txt"

tab = astropy.table.Table.read(loc, format='ascii')
```

[74]:
```
tab.add_column(np.arange(len(tab))+1, index=0, name='index')
```

[ ]:
```

```

[76]:
```
monr2 = tab[tab['Name'] == 'Mon R2']
```

[10]:
```
monr2_coords = coords_from_Luhman_table(monr2)
```

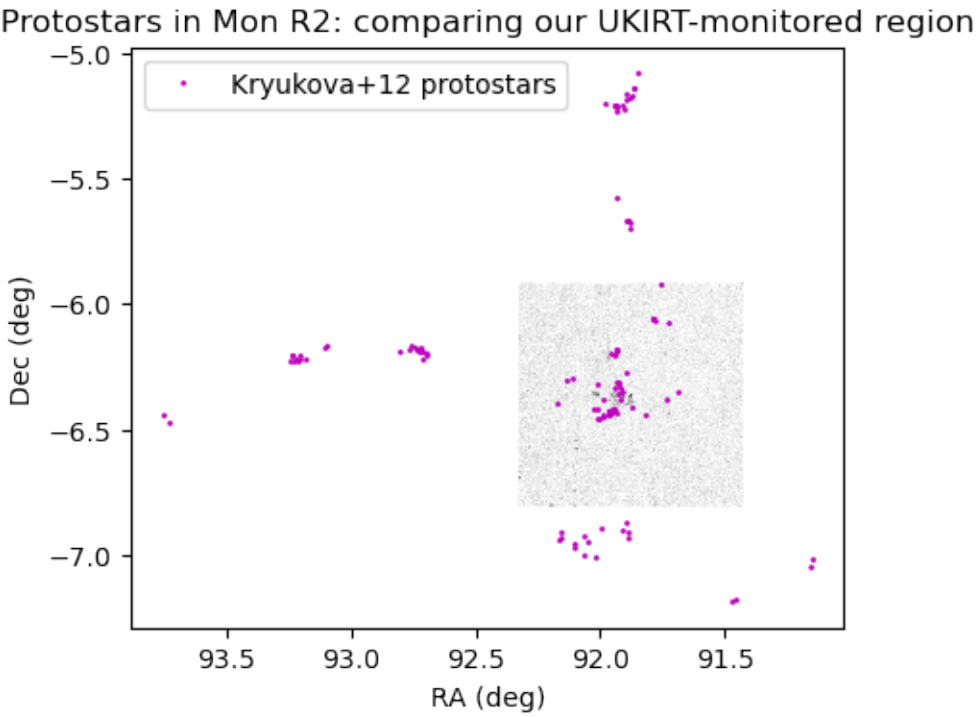
[11]:
```
mon_spread = spreadsheet.load_wserv_v2(11)
mon_q = quality_classes.load_q(11)
mon_dat = photometry.group_wserv_v2(photometry.load_wserv_v2(11))
```

```
Loading WSERV11 photometry data… DONE (elapsed time: 0.31s)
```

[25]:
```
fig, ax = plt.subplots(1, dpi=100)

plt.plot(np.degrees(mon_spread['median']['RA']), np.
 ↪degrees(mon_spread['median']['DEC']), 'k,', alpha=0.025)
plt.plot(monr2_coords.ra, monr2_coords.dec, 'm.', ms=2, label="Kryukova+12␣
 ↪protostars") # , scalex=False, scaley=False)
```

```
ax.set_aspect(1/np.cos(np.radians(-6.4)))
ax.invert_xaxis()
ax.set_xlabel("RA (deg)")
ax.set_ylabel("Dec (deg)")

ax.legend()

ax.set_title("Protostars in Mon R2: comparing our UKIRT-monitored region")
```

[25]: Text(0.5, 1.0, 'Protostars in Mon R2: comparing our UKIRT-monitored region')



[26]:
```
mon_sm = mon_spread["median"]
mon_spreadsheet_coordinates = SkyCoord(
    ra=mon_sm["RA"].values * u.rad, dec=mon_sm["DEC"].values * u.rad
)
```

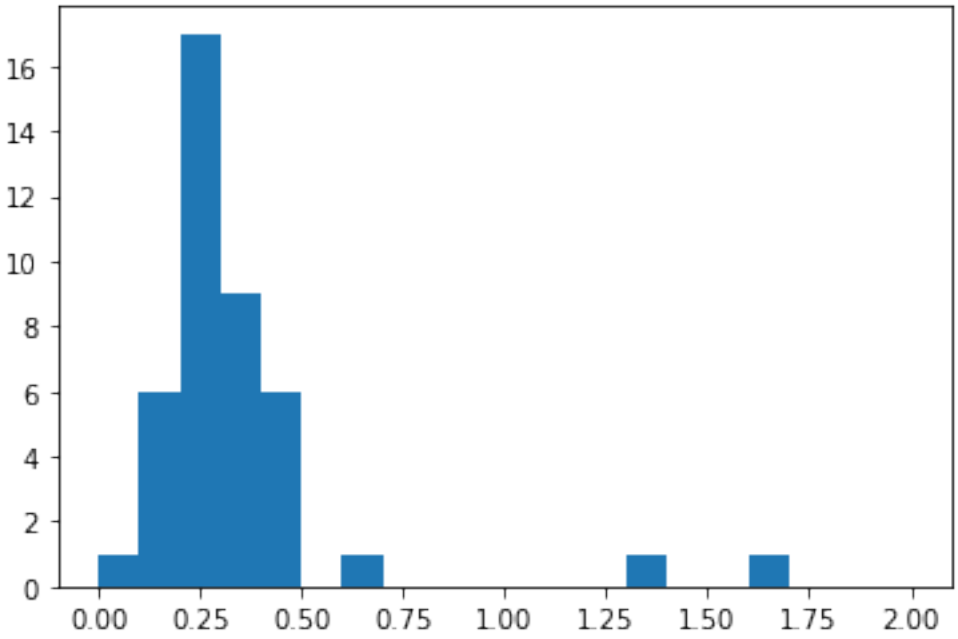
[28]:
```
idx, d2d, d3d = monr2_coords.match_to_catalog_sky(mon_spreadsheet_coordinates)
```

[ ]:
```

```

[87]:
```
sokol_idx, sokol_d2d, sokol_d3d = sokol_proto_coords.
↪match_to_catalog_sky(mon_spreadsheet_coordinates)
```

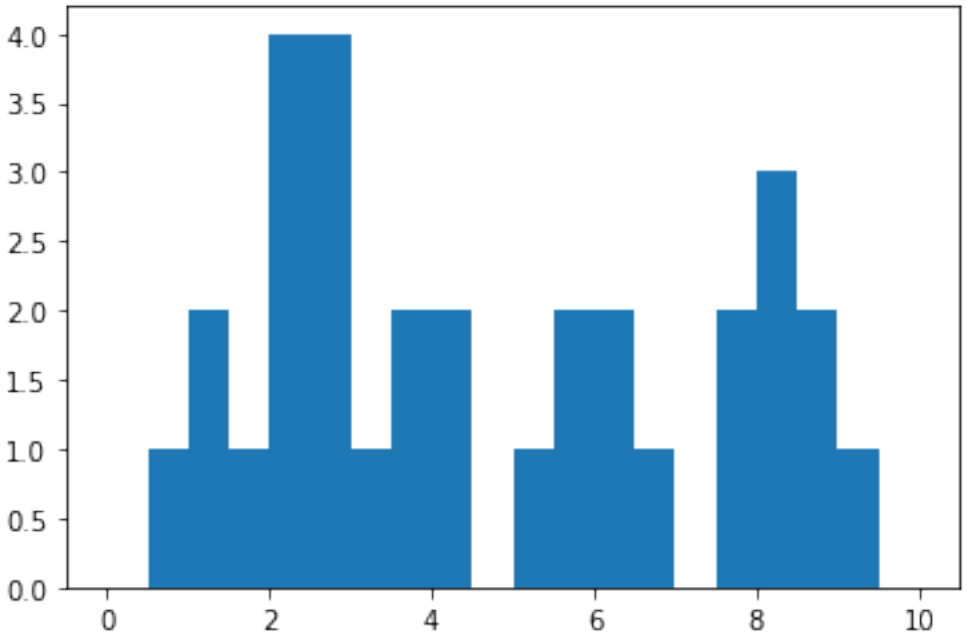
[31]:
```
plt.subplots(1)
# plt.hist(d2d.to(u.arcsec).value, bins=50, range=(0,1), label='All NGC 1333␣
 ↪members')
plt.hist(d2d.to(u.arcsec).value, range=(0, 2), bins=20)
```

[31]:
```
(array([ 1.,  6., 17.,  9.,  6.,  0.,  1.,  0.,  0.,  0.,  0.,  0.,  0.,
         1.,  0.,  0.,  1.,  0.,  0.,  0.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
        1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. ]),
 <BarContainer object of 20 artists>)
```



[89]:
```
plt.subplots(1)
# plt.hist(d2d.to(u.arcsec).value, bins=50, range=(0,1), label='All NGC 1333␣
 ↪members')
plt.hist(sokol_d2d.to(u.arcsec).value, range=(0, 10), bins=20)

# looks like Sokol isn't helpful.
```

[89]:
```
(array([0., 1., 2., 1., 4., 4., 1., 2., 2., 0., 1., 2., 2., 1., 0., 2., 3.,
        2., 1., 0.]),
 array([ 0. ,  0.5,  1. ,  1.5,  2. ,  2.5,  3. ,  3.5,  4. ,  4.5,  5. ,
         5.5,  6. ,  6.5,  7. ,  7.5,  8. ,  8.5,  9. ,  9.5, 10. ]),
 <BarContainer object of 20 artists>)
```



[77]:
```
max_sep = 0.5 * u.arcsec
sep_constraint = d2d < max_sep

# We're going to compute
# (a) all matched IC348 members, for quality control / comparisons, and
# (b) just the brown dwarfs

matches = mon_sm.iloc[idx[sep_constraint]]
matched = monr2[sep_constraint]
```

[78]:
```
matches
```

[78]:

SOURCEID	MEANMJDOBS	RA	DEC	JMHPNT	JMHPNTERR	\
45195940882456	57498.233804	1.600250	-0.110830	0.886533	0.155289	
45195940912937	57499.252958	1.600975	-0.106073	NaN	NaN	
45195940900509	57496.733632	1.600996	-0.111342	1.060341	0.270495	
45195940912813	57499.252958	1.601489	-0.103401	NaN	NaN	
45195940910910	57499.252958	1.601908	-0.105830	1.563782	0.130744	
45195940910989	57498.742648	1.601973	-0.105694	1.826439	0.032666	
45195940910935	57498.742648	1.602051	-0.105806	1.449108	0.032325	
45195940901643	57500.745944	1.602572	-0.112456	NaN	NaN	
45195940899279	57498.742648	1.603485	-0.111920	3.423397	0.191917	
45195940863213	57498.739607	1.603895	-0.109514	1.266585	0.036751	
45195940888875	57496.232847	1.604248	-0.111373	1.361532	0.021452	
45195940863458	57498.739607	1.604339	-0.110545	1.855111	0.041041	

```
45195940863492  57498.739607  1.604401 -0.110233  2.969787   0.046311
45195940867046  57498.229348  1.604465 -0.110267       NaN        NaN
45195940867056  57477.732180  1.604489 -0.110231       NaN        NaN
45195940863573  57498.739607  1.604495 -0.107997  1.982880   0.034785
45195940891508  57497.231408  1.604517 -0.112263       NaN        NaN
45195940863570  57498.739607  1.604519 -0.107852  1.945163   0.190058
45195940863588  57498.739607  1.604540 -0.107931       NaN        NaN
45195940889449  57498.741101  1.604623 -0.111990  2.182735   0.027175
45195940866107  57492.238463  1.604629 -0.108297       NaN        NaN
45195940889471  57498.741101  1.604751 -0.112028  2.609900   0.041100
45195940892275  57498.741101  1.604804 -0.112144  2.106922   0.149349
45195940889555  57498.741101  1.604830 -0.112175  1.449080   0.021364
45195940891442  57497.731140  1.604843 -0.112027       NaN        NaN
45195940889591  57498.741101  1.604888 -0.112260  2.571402   0.030502
45195940889645  57498.741101  1.605000 -0.112362  1.741518   0.023036
45195940889580  57498.741101  1.605063 -0.112224  1.499331   0.025502
45195940892289  57499.251331  1.605076 -0.112193  1.488379   0.150334
45195940889661  57498.741101  1.605083 -0.112393  1.797954   0.045820
45195940889579  57498.741101  1.605108 -0.112184  1.931623   0.022621
45195940888862  57496.232847  1.605434 -0.111387  2.321815   0.037159
45195940889680  57498.741101  1.605539 -0.112483  1.460678   0.036436
45195940891595  57498.741101  1.605724 -0.112675       NaN        NaN
45195940889479  57498.741101  1.605875 -0.112049  2.336576   0.084180
45195940864698  57498.739607  1.605918 -0.110260  2.309396   0.140451
45195940889526  57498.741101  1.606131 -0.112101  2.787045   0.112865
45195940878929  57499.254763  1.608100 -0.110010  2.682369   0.049414
45195940903063  57498.742648  1.608676 -0.111634  3.095667   0.135277

                   HMKPNT  HMKPNTERR  JAPERMAG3  JAPERMAG3ERR  HAPERMAG3  … \
SOURCEID                                                                  …
45195940882456  1.323406   0.111682  18.758604      0.111104  17.930027  …
45195940912937       NaN        NaN        NaN           NaN        NaN  …
45195940900509  0.971643   0.212837  19.749649      0.198471  18.653749  …
45195940912813  2.481479   0.214292        NaN           NaN  18.698359  …
45195940910910  2.038729   0.054970  18.445070      0.114169  17.111919  …
45195940910989  1.534174   0.022775  16.713936      0.024265  14.903908  …
45195940910935  1.232316   0.023905  16.613245      0.022552  15.179688  …
45195940901643  1.954359   0.151620        NaN           NaN  18.597343  …
45195940899279  2.615606   0.040737  19.519833      0.169511  16.842949  …
45195940863213  0.971352   0.027110  17.036081      0.026212  15.765283  …
45195940888875  0.874716   0.021146  14.000462      0.009961  12.639738  …
45195940863458  1.182107   0.024779  17.345442      0.031870  15.485445  …
45195940863492  1.922712   0.021889  17.514849      0.038662  14.564892  …
45195940867046       NaN        NaN        NaN           NaN        NaN  …
45195940867056       NaN        NaN        NaN           NaN        NaN  …
45195940863573  2.006802   0.022621  16.995289      0.025860  15.009662  …
45195940891508  1.627364   0.249871        NaN           NaN  19.056019  …
```

45195940863570	1.746381	0.081322	19.603325	0.159890	17.760769	…
45195940863588	2.190234	0.082431	NaN	NaN	17.856522	…
45195940889449	1.806528	0.021557	16.303234	0.018134	14.124305	…
45195940866107	0.910697	0.241973	NaN	NaN	18.810623	…
45195940889471	1.837265	0.022061	17.220024	0.033961	14.603916	…
45195940892275	2.189340	0.047468	18.978409	0.131218	17.020584	…
45195940889555	1.217927	0.021096	13.789535	0.009857	12.357391	…
45195940891442	1.684175	0.160758	NaN	NaN	18.711555	…
45195940889591	2.156978	0.021518	16.591370	0.023261	14.022272	…
45195940889645	1.629931	0.021316	15.339421	0.012273	13.605362	…
45195940889580	1.172744	0.022089	15.959610	0.015367	14.467332	…
45195940892289	2.254236	0.052153	18.515377	0.125953	17.095711	…
45195940889661	1.560502	0.025065	17.323109	0.036514	15.534996	…
45195940889579	1.508393	0.021226	15.192266	0.011830	13.269271	…
45195940888862	1.507772	0.022256	16.974998	0.029779	14.638508	…
45195940889680	1.442612	0.024685	16.907845	0.026644	15.442160	…
45195940891595	1.534107	0.194091	NaN	NaN	18.721094	…
45195940889479	1.706914	0.026948	18.251642	0.077547	15.918461	…
45195940864698	1.970806	0.039739	18.956854	0.121728	16.771675	…
45195940889526	2.317344	0.026159	18.459068	0.100457	15.824811	…
45195940878929	2.354242	0.022483	17.764093	0.040967	15.106825	…
45195940903063	2.308733	0.026003	18.854736	0.119562	15.765606	…

SOURCEID	KAPERMAG3	KAPERMAG3ERR	JPPERRBITS	HPPERRBITS	KPPERRBITS	\
45195940882456	16.581913	0.049790	0.0	0.0	0.0	
45195940912937	17.226006	0.092387	NaN	NaN	0.0	
45195940900509	17.692110	0.121292	0.0	0.0	0.0	
45195940912813	16.302380	0.040579	NaN	0.0	0.0	
45195940910910	14.990565	0.016126	0.0	0.0	0.0	
45195940910989	13.374398	0.010095	0.0	0.0	0.0	
45195940910935	13.949826	0.010929	0.0	16.0	16.0	
45195940901643	16.690296	0.049481	NaN	0.0	0.0	
45195940899279	14.207936	0.011673	0.0	0.0	0.0	
45195940863213	14.803169	0.013586	0.0	0.0	0.0	
45195940888875	11.765326	0.009416	0.0	16.0	16.0	
45195940863458	14.306738	0.011485	0.0	0.0	0.0	
45195940863492	12.647103	0.009601	0.0	0.0	0.0	
45195940867046	16.679594	0.049771	NaN	NaN	0.0	
45195940867056	18.107368	0.156098	NaN	NaN	0.0	
45195940863573	12.967777	0.009738	0.0	0.0	0.0	
45195940891508	17.462837	0.101687	NaN	0.0	0.0	
45195940863570	16.004986	0.028854	0.0	0.0	0.0	
45195940863588	15.543564	0.020835	NaN	0.0	0.0	
45195940889449	12.323475	0.009511	16.0	16.0	16.0	
45195940866107	17.962614	0.152970	NaN	0.0	0.0	
45195940889471	12.762114	0.009672	0.0	0.0	0.0	

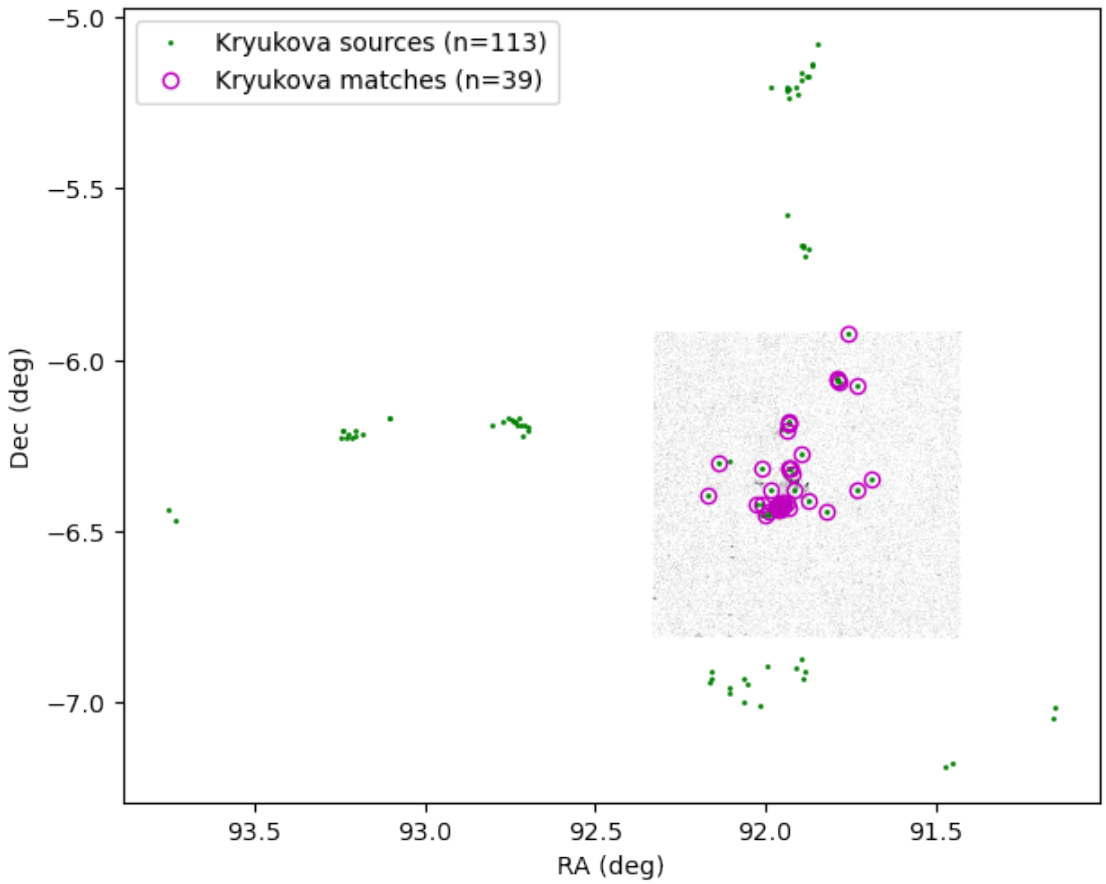
45195940892275	14.782052	0.013977	0.0	16.0	16.0
45195940889555	11.152027	0.009363	0.0	16.0	16.0
45195940891442	16.917933	0.063729	NaN	0.0	0.0
45195940889591	11.892399	0.009433	0.0	0.0	0.0
45195940889645	11.974230	0.009444	16.0	16.0	16.0
45195940889580	13.285858	0.009960	0.0	16.0	16.0
45195940892289	14.846551	0.014224	0.0	16.0	16.0
45195940889661	13.961138	0.010847	16.0	16.0	16.0
45195940889579	11.764299	0.009418	16.0	16.0	16.0
45195940888862	13.148381	0.009855	0.0	0.0	0.0
45195940889680	13.965570	0.010880	0.0	0.0	0.0
45195940891595	17.169634	0.081034	NaN	0.0	0.0
45195940889479	14.225718	0.011368	0.0	0.0	0.0
45195940864698	14.807493	0.013611	0.0	0.0	0.0
45195940889526	13.456480	0.010150	0.0	16.0	16.0
45195940878929	12.678046	0.009605	0.0	16.0	16.0
45195940903063	13.449727	0.010086	0.0	0.0	0.0

SOURCEID	MERGEDCLASS	PSTAR	JGRADE	HGRADE	KGRADE
45195940882456	1.0	0.003067	0.983819	0.970488	0.969743
45195940912937	1.0	0.050000	0.983193	0.966970	0.970359
45195940900509	-1.0	0.900000	0.983277	0.968912	0.970724
45195940912813	-1.0	0.486486	0.981697	0.964571	0.969088
45195940910910	1.0	0.000171	0.981697	0.964571	0.969088
45195940910989	1.0	0.052478	0.981773	0.964387	0.969016
45195940910935	1.0	0.000171	0.981773	0.964387	0.969016
45195940901643	1.0	0.050000	0.981773	0.965123	0.969136
45195940899279	1.0	0.003067	0.981773	0.964387	0.969016
45195940863213	1.0	0.000171	0.977797	0.968273	0.970369
45195940888875	-1.0	0.999657	0.981343	0.968274	0.966292
45195940863458	1.0	0.000171	0.977797	0.968273	0.970369
45195940863492	1.0	0.000171	0.977797	0.968273	0.970369
45195940867046	1.0	0.050000	0.977891	0.969376	0.970948
45195940867056	1.0	0.050000	0.990596	0.980526	0.985736
45195940863573	1.0	0.000171	0.977797	0.968273	0.970369
45195940891508	-1.0	0.900000	0.983364	0.972671	0.969005
45195940863570	1.0	0.003067	0.977797	0.968273	0.970369
45195940863588	1.0	0.003067	0.977797	0.968273	0.970369
45195940889449	1.0	0.003067	0.981378	0.970207	0.966833
45195940866107	1.0	0.050000	0.983766	0.975799	0.974920
45195940889471	1.0	0.052478	0.981378	0.970207	0.966833
45195940892275	1.0	0.003067	0.981378	0.970207	0.966833
45195940889555	1.0	0.052478	0.981378	0.970207	0.966833
45195940891442	1.0	0.050000	0.983287	0.971912	0.968877
45195940889591	-1.0	0.944606	0.981378	0.970207	0.966833
45195940889645	-1.0	0.999657	0.981378	0.970207	0.966833

```
45195940889580          1.0  0.000171  0.981378  0.970207  0.966833
45195940892289          1.0  0.050000  0.981413  0.971393  0.967374
45195940889661          1.0  0.000171  0.981378  0.970207  0.966833
45195940889579          1.0  0.052478  0.981378  0.970207  0.966833
45195940888862         -1.0  0.999657  0.981343  0.968274  0.966292
45195940889680         -1.0  0.999657  0.981378  0.970207  0.966833
45195940891595          1.0  0.050000  0.983287  0.972603  0.968877
45195940889479         -1.0  0.999657  0.981378  0.970207  0.966833
45195940864698         -1.0  0.993865  0.977797  0.968273  0.970369
45195940889526          1.0  0.000171  0.981378  0.970207  0.966833
45195940878929          1.0  0.000171  0.983819  0.969900  0.968278
45195940903063          1.0  0.000171  0.981773  0.964387  0.969016

[39 rows x 21 columns]
```

[41]:
```
fig, ax = plt.subplots(1, figsize=(7,7), dpi=100)

plt.plot(np.degrees(mon_sm['RA']), np.degrees(mon_sm['DEC']), 'k,', alpha=0.03)
plt.plot(monr2_coords.ra, monr2_coords.dec, 'g.', ms=2, label=f'Kryukova␣
 ↪sources (n={len(monr2)})')
plt.plot(np.degrees(matches['RA']), np.degrees(matches['DEC']), 'mo',␣
 ↪markerfacecolor='None', label=f'Kryukova matches (n={len(matches)})')

ax.set_aspect(1/np.cos(np.radians(-6.4)))
ax.invert_xaxis()
ax.set_xlabel("RA (deg)")
ax.set_ylabel("Dec (deg)")

ax.legend()
```

[41]:
```
<matplotlib.legend.Legend at 0x7fcb044eabd0>
```



[79]:
```
matched
```

[79]:
```
<Table length=39>
index  Name  f_Name  RAh   RAm   …  24mag  e_24mag f_24mag  alpha     logL
                     h     min   …   mag     mag                   dex(Lsun)
int64 str13   str1  int64 int64 … float64 float64   str1   float64  float64
----- ------ ------ ----- ----- … ------- ------- ------- ------- ---------
  620 Mon R2     --     6     6 …    7.74    0.03      --    0.35     -1.32
  621 Mon R2     --     6     6 …     4.9    0.01      --     0.9     -0.02
  622 Mon R2     --     6     6 …    7.39    0.04      --    1.27     -0.93
  623 Mon R2     --     6     7 …     7.3    0.02      --    0.31     -1.18
  624 Mon R2     --     6     7 …    2.04    0.01      --     2.0     1.378
  625 Mon R2     --     6     7 …    4.82    0.01      --    0.46     -0.07
  627 Mon R2     --     6     7 …    4.44    0.01      --    0.96     0.169
  628 Mon R2     --     6     7 …    6.83    0.02      --    1.03     -0.95
  632 Mon R2     --     6     7 …    3.46    0.02      --   -0.03     0.553
  …    …     …    …   … …     …       …      …       …       …
  678 Mon R2     --     6     7 …    4.54    0.02      --    0.79     0.051
  679 Mon R2     --     6     7 …    3.68    0.01      --   -0.11     0.572
```

```
 682 Mon R2     --     6     7 …    5.17    0.02      --    0.32     -0.25
 683 Mon R2     --     6     7 …    6.06    0.02      --    0.12     -0.47
 686 Mon R2     --     6     8 …    7.29    0.02      --    1.18     -1.14
 687 Mon R2     --     6     8 …    6.33    0.02      --   -0.21     -0.41
 689 Mon R2     --     6     8 …    6.72    0.03      --   -0.03     -0.76
 691 Mon R2     --     6     8 …    5.36    0.01      --   -0.14     -0.11
 698 Mon R2     --     6     8 …    2.14    0.01      --    0.56      1.03
 702 Mon R2     --     6     8 …    3.24    0.01      --    0.46     0.491
```

[45]:
```
# also checking Sokol

sokol_loc = "/Users/tsrice/Documents/Variability_Project_2020/wuvars/data/
 ↪auxiliary_catalogs/c2d_MonR2/Sokol_2019_Table1_sty3107_supplemental_table.
 ↪txt"
sokol = astropy.table.Table.read(sokol_loc, format='ascii.latex')
```

[48]:
```
sokol_protostars = sokol[sokol['Protostar?'] == 'Y']
```

[ ]:
```

```

[54]:
```
sokol_proto_coords = SkyCoord(ra=sokol_protostars['RA'],␣
 ↪dec=sokol_protostars['Dec'], unit=('hour', 'deg'))
```

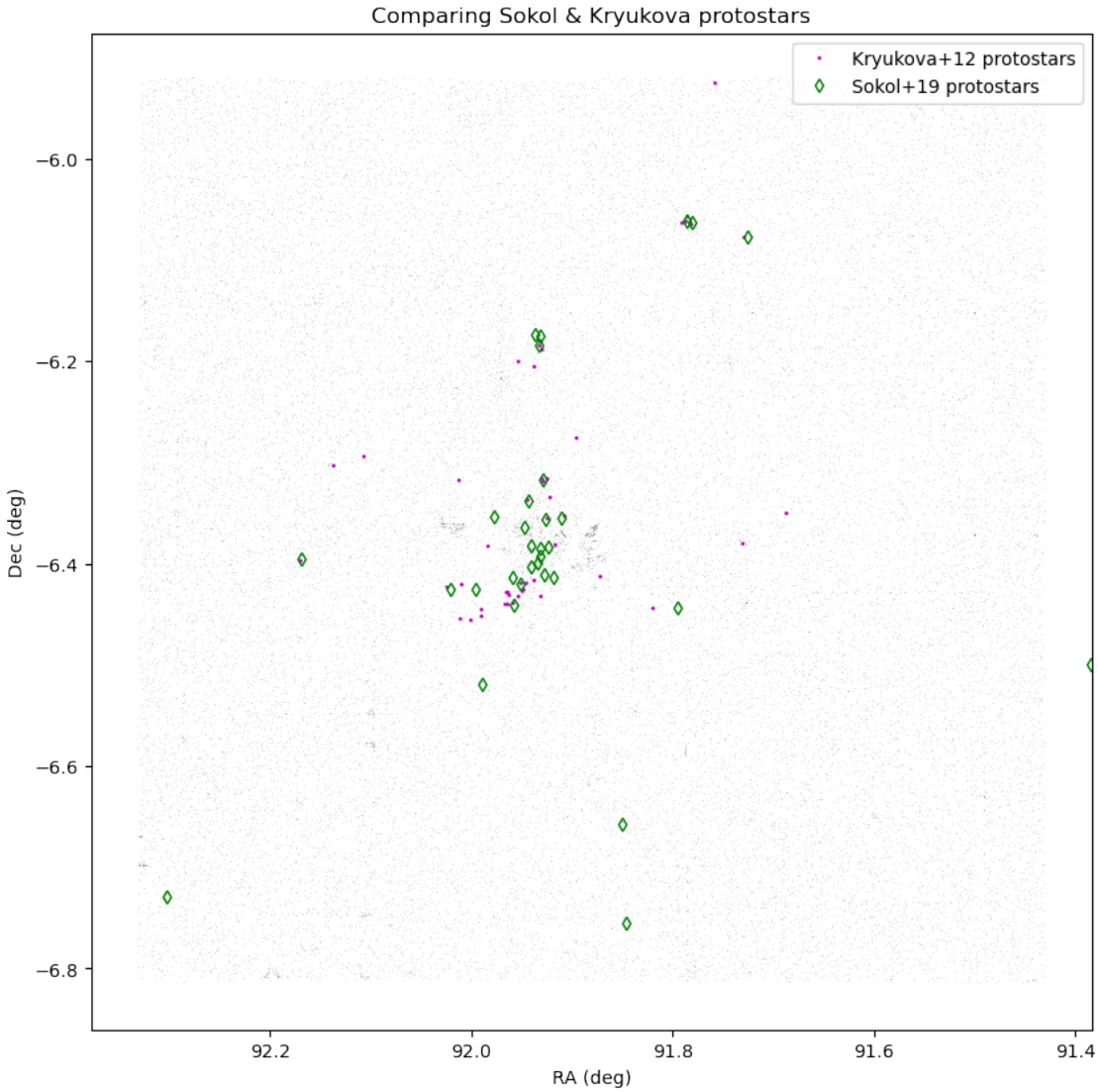
[55]:
```
sokol_proto_coords
```

[55]:
```
<SkyCoord (ICRS): (ra, dec) in deg
    [(91.43583333, -7.16241667), (91.28458333, -7.24363889),
     (91.45416667, -7.17811111), (91.47458333, -7.18783333),
     (91.15375   , -7.04694444), (92.15833333, -6.92944444),
     (92.22541667, -7.02733333), (92.10416667, -6.96169444),
     (91.88333333, -6.90930556), (91.90791667, -6.90502778),
     (92.13875   , -6.88388889), (92.01625   , -7.01002778),
     (92.17583333, -6.96083333), (91.995     , -7.07586111),
     (92.05375   , -6.94516667), (91.88791667, -6.92211111),
     (92.15666667, -6.90916667), (92.13708333, -6.98486111),
     (91.84625   , -6.75444444), (92.3025    , -6.7295    ),
     (91.24      , -6.58766667), (91.38458333, -6.49919444),
     (93.72875   , -6.36716667), (93.69291667, -6.37008333),
     (93.82208333, -6.41369444), (91.94125   , -6.38255556),
     (91.93166667, -6.38466667), (91.93208333, -6.39230556),
     (91.93375   , -6.39925   ), (91.9475    , -6.36366667),
     (91.94041667, -6.40283333), (91.94333333, -6.33852778),
     (91.92583333, -6.35577778), (91.91833333, -6.41313889),
     (91.91041667, -6.35522222), (91.95125   , -6.41991667),
     (91.78625   , -6.06133333), (91.95875   , -6.41394444),
     (91.92375   , -6.38327778), (91.92875   , -6.31730556),
     (91.85083333, -6.65702778), (91.98916667, -6.51908333),
```

```
(91.99625   , -6.42491667), (91.78083333, -6.06327778),
(91.795     , -6.443     ), (91.95833333, -6.44075   ),
(91.9275    , -6.4105    ), (91.97791667, -6.35352778),
(92.02041667, -6.42463889), (91.93291667, -6.18355556),
(91.93125   , -6.17536111), (91.93708333, -6.17369444),
(92.16833333, -6.39511111), (91.72583333, -6.07772222),
(93.2075    , -6.22588889), (93.205     , -6.20575   ),
(93.1875    , -6.21741667), (92.70916667, -6.19563889),
(92.71791667, -6.19230556), (92.77041667, -6.18425   ),
(92.69458333, -6.20661111), (92.74708333, -6.18258333),
(92.67708333, -6.21244444), (92.68708333, -6.20925   ),
(92.80416667, -6.19508333), (92.7225    , -6.16744444),
(92.72208333, -6.17730556), (92.72666667, -6.18758333),
(92.74666667, -6.17522222), (91.76916667, -5.62433333),
(91.79875   , -5.78405556), (91.89      , -5.66908333),
(91.935     , -5.57855556), (91.79375   , -5.60544444),
(92.17291667, -5.71797222), (91.89125   , -5.6995    ),
(91.78375   , -5.62461111), (91.93458333, -5.56975   ),
(91.93541667, -5.57683333), (91.93958333, -5.511     ),
(91.87625   , -5.17127778), (91.96958333, -5.26669444),
(91.85916667, -5.14044444), (91.99958333, -5.13211111),
(91.95      , -5.21863889), (91.84833333, -5.13488889)]>
```

[67]:
```
fig, ax = plt.subplots(1, figsize=(10,10), dpi=100)

plt.plot(np.degrees(mon_spread['median']['RA']), np.
 ↪degrees(mon_spread['median']['DEC']), 'k,', alpha=0.1)
ax.invert_xaxis()

ax.plot(monr2_coords.ra, monr2_coords.dec, 'm.', ms=2, label="Kryukova+12␣
 ↪protostars", scalex=False, scaley=False)
ax.plot(sokol_proto_coords.ra, sokol_proto_coords.dec, 'gd', ms=5, mfc='None',␣
 ↪label="Sokol+19 protostars", scalex=False, scaley=False)

ax.set_aspect(1/np.cos(np.radians(-6.4)))
ax.set_xlabel("RA (deg)")
ax.set_ylabel("Dec (deg)")

ax.legend()

ax.set_title("Comparing Sokol & Kryukova protostars")
```

[67]:
```
Text(0.5, 1.0, 'Comparing Sokol & Kryukova protostars')
```



[68]:
```
kryukova_match_sids = matches.index
```

[69]:
```
kryukova_match_sids
```

[69]:
```
Int64Index([45195940882456, 45195940912937, 45195940900509, 45195940912813,
            45195940910910, 45195940910989, 45195940910935, 45195940901643,
            45195940899279, 45195940863213, 45195940888875, 45195940863458,
            45195940863492, 45195940867046, 45195940867056, 45195940863573,
            45195940891508, 45195940863570, 45195940863588, 45195940889449,
            45195940866107, 45195940889471, 45195940892275, 45195940889555,
            45195940891442, 45195940889591, 45195940889645, 45195940889580,
            45195940892289, 45195940889661, 45195940889579, 45195940888862,
            45195940889680, 45195940891595, 45195940889479, 45195940864698,
```

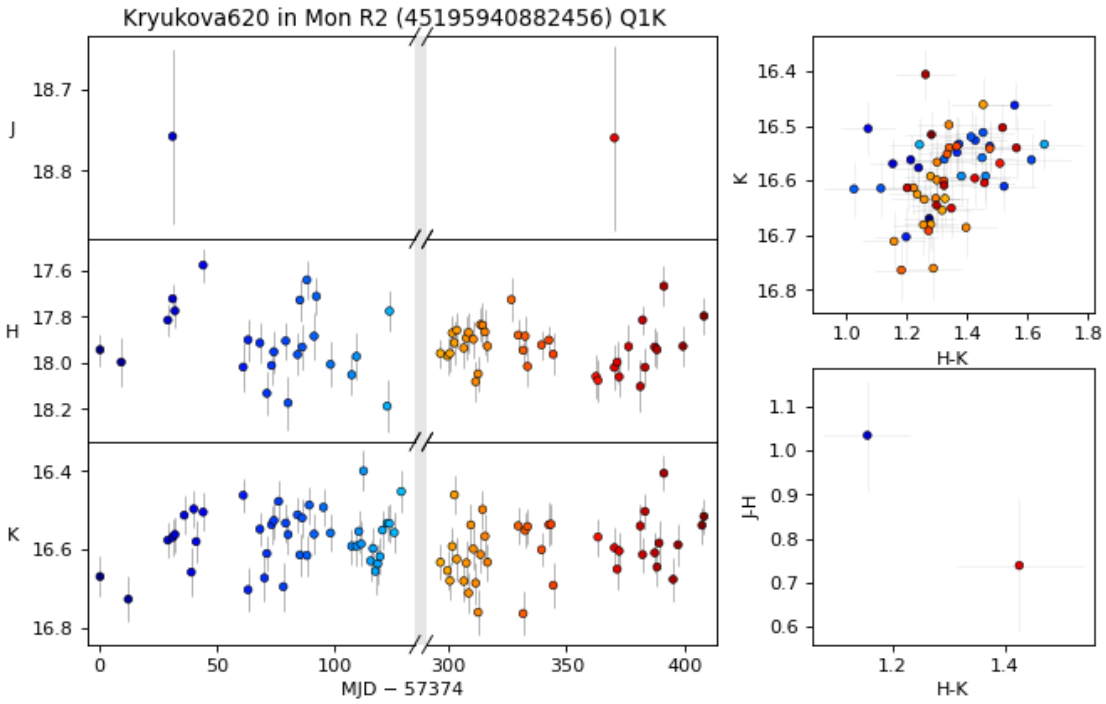
```
 45195940889526, 45195940878929, 45195940903063],
dtype='int64', name='SOURCEID')
```

[70]:
```
from wuvars.plotting.lightcurve import monr2_simple_lc_scatter_brokenaxes
```

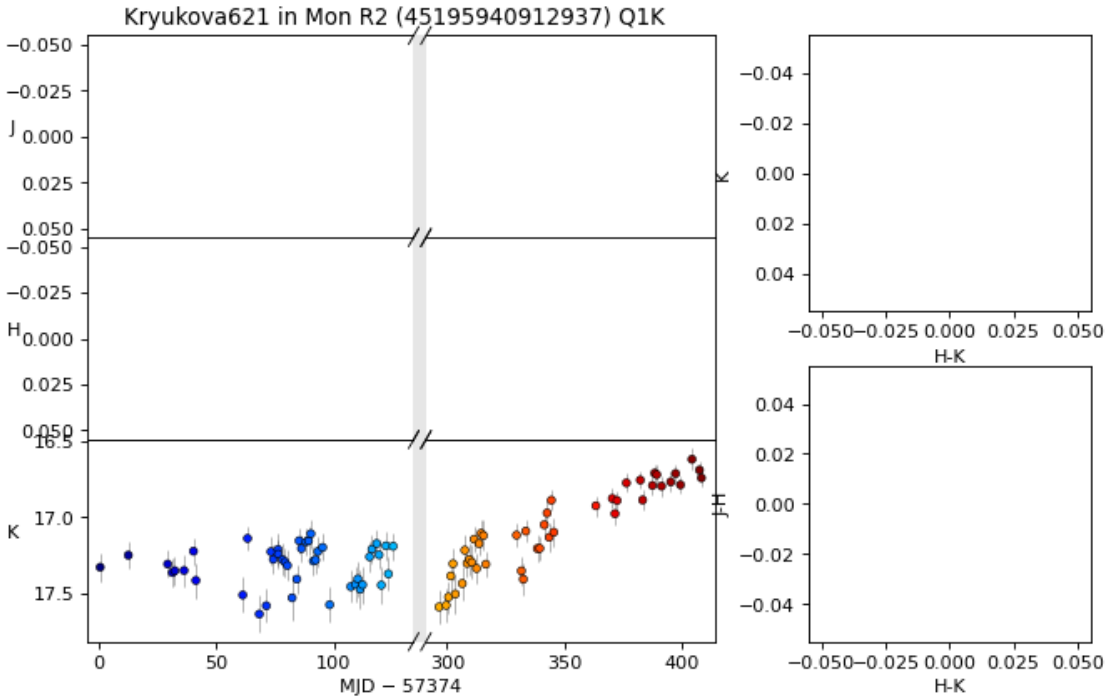
[71]:
```
def q_string(sid, spread, qualityset):

    q = qualityset

    if sid in spread[q.q2].index:
        return "2"

    elif sid in spread[q.q1_j | q.q1_h | q.q1_k].index:
        return_string = "1"

        if sid in spread[q.q1_j].index:
            return_string += "J"
        if sid in spread[q.q1_h].index:
            return_string += "H"
        if sid in spread[q.q1_k].index:
            return_string += "K"

        return return_string

    elif sid in spread[q.q0].index:
        return "0"

    else:
        return "-1"
```

[85]:
```
#     data_quality_text = f"Q{q_string(sid, spread, q):4s}  "


for i, sid in enumerate(kryukova_match_sids):

    print(i, sid)
    fig_lc = monr2_simple_lc_scatter_brokenaxes(mon_dat, sid, cmap='jet')
    fig_lc.ax_j.set_title(f"Kryukova{matched['index'][i]} in Mon R2 ({sid})␣
↪Q{q_string(sid, mon_spread, mon_q):4s}")


    plt.show()
```

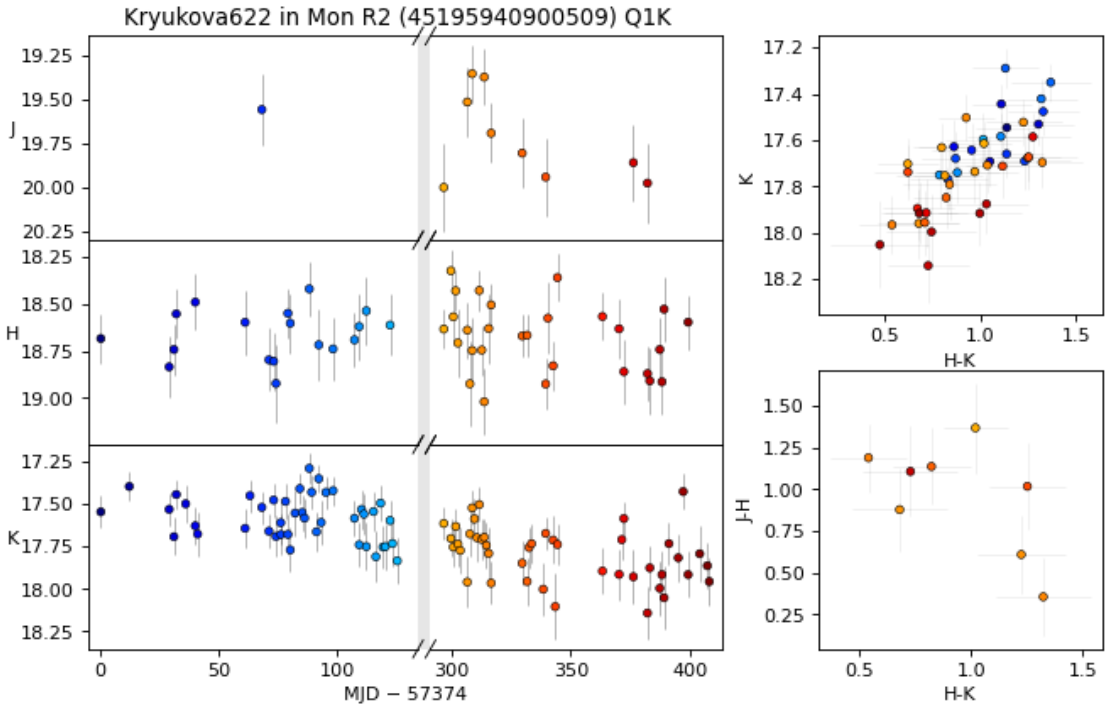
```
0 45195940882456
```



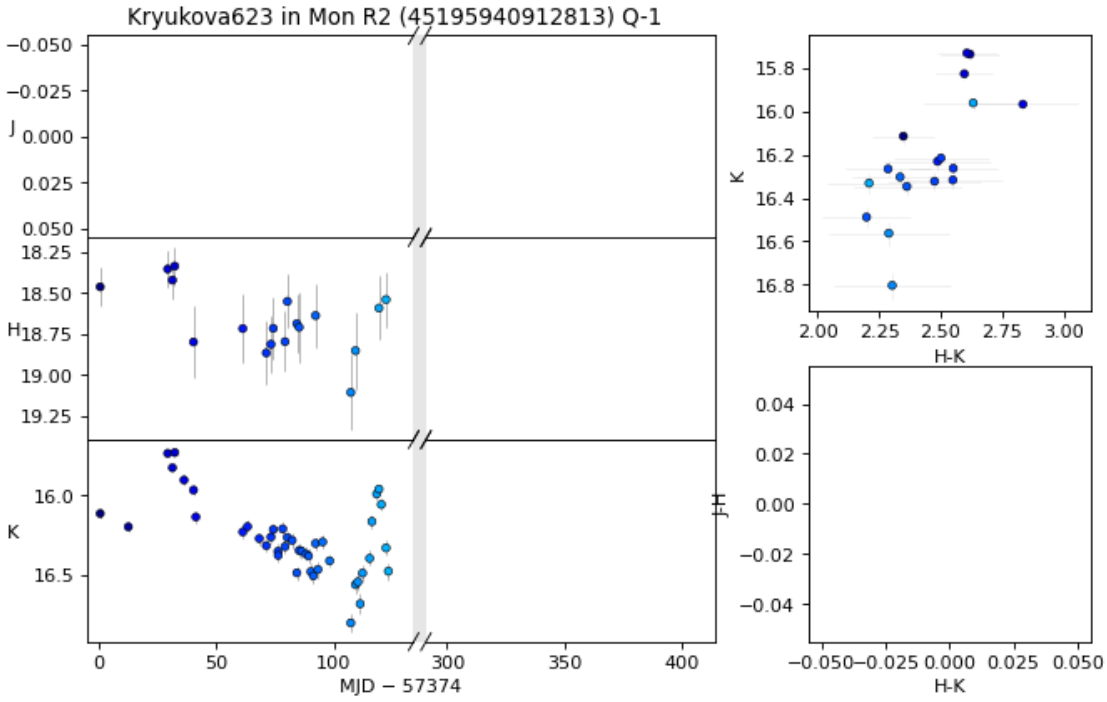

```
1 45195940912937

/Users/tsrice/opt/anaconda3/lib/python3.7/site-
packages/numpy/core/_asarray.py:102: UserWarning: Warning: converting a masked
element to nan.
  return array(a, dtype, copy=False, order=order)
```



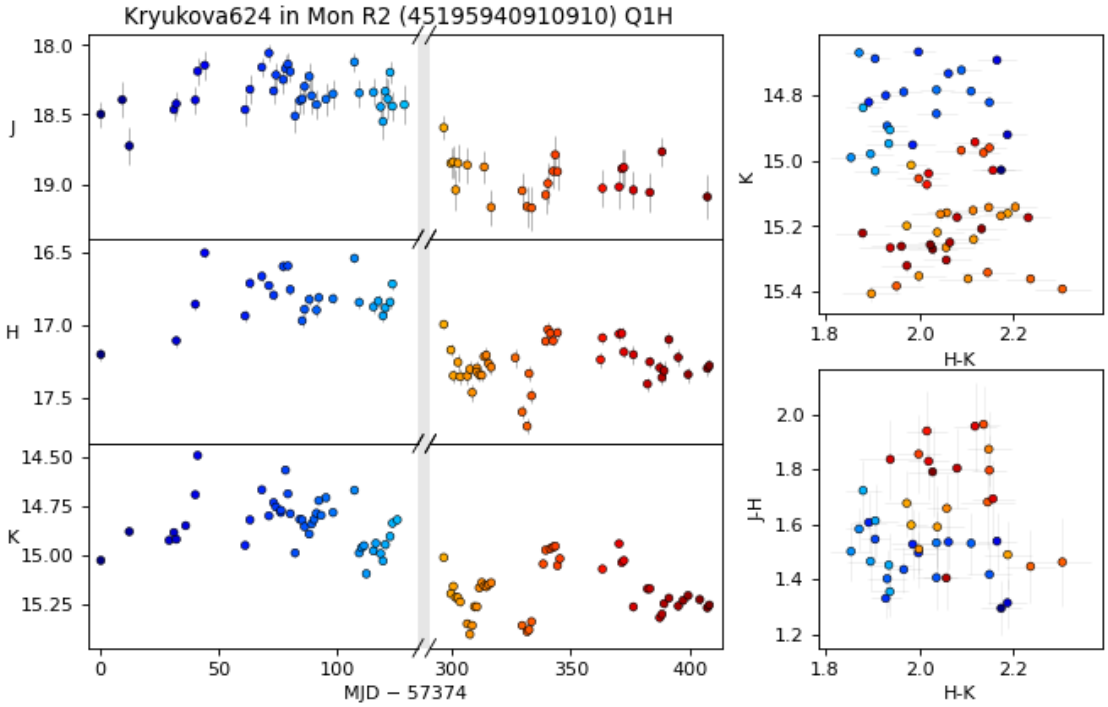
2 45195940900509



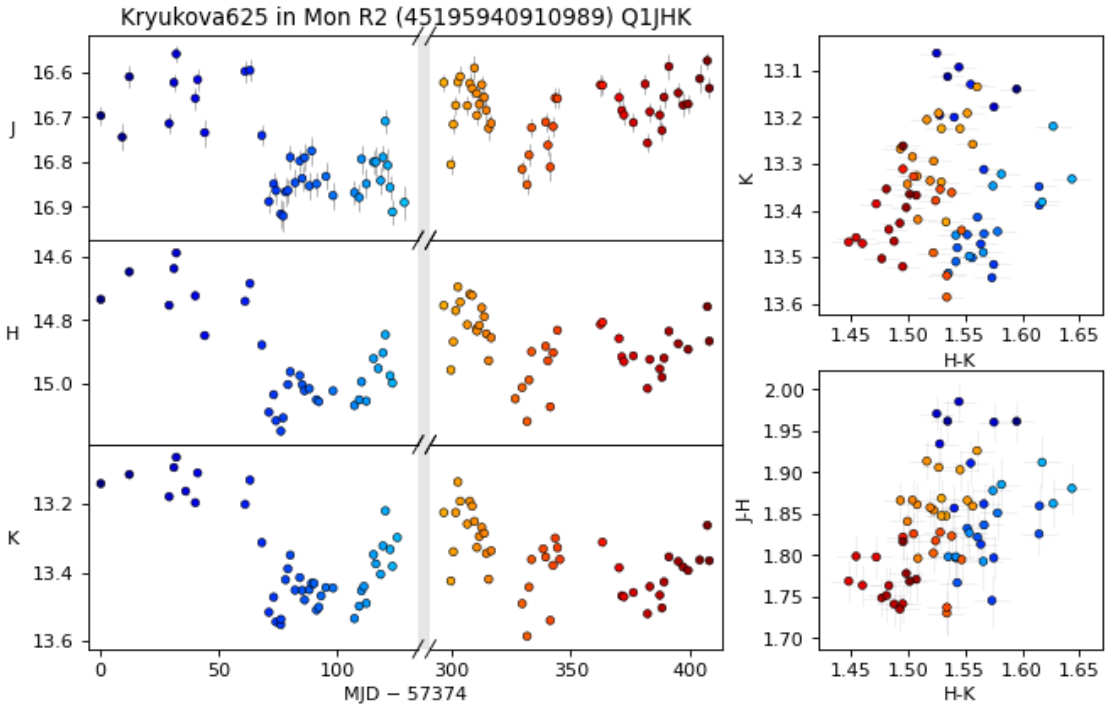
3 45195940912813



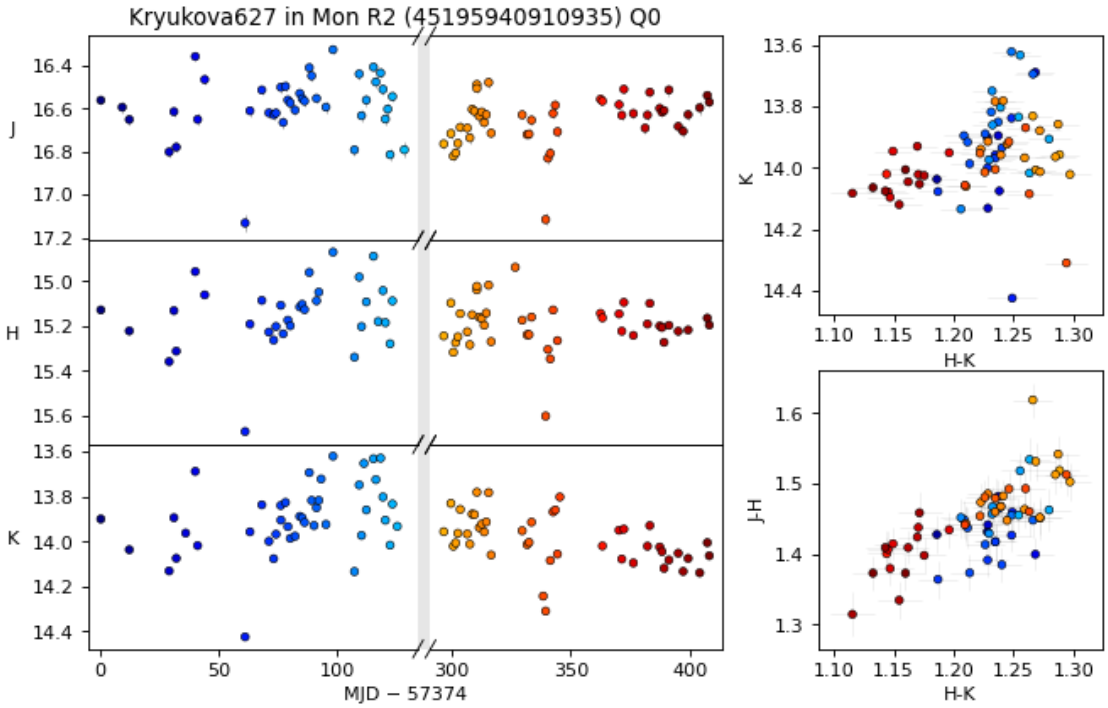
4 45195940910910



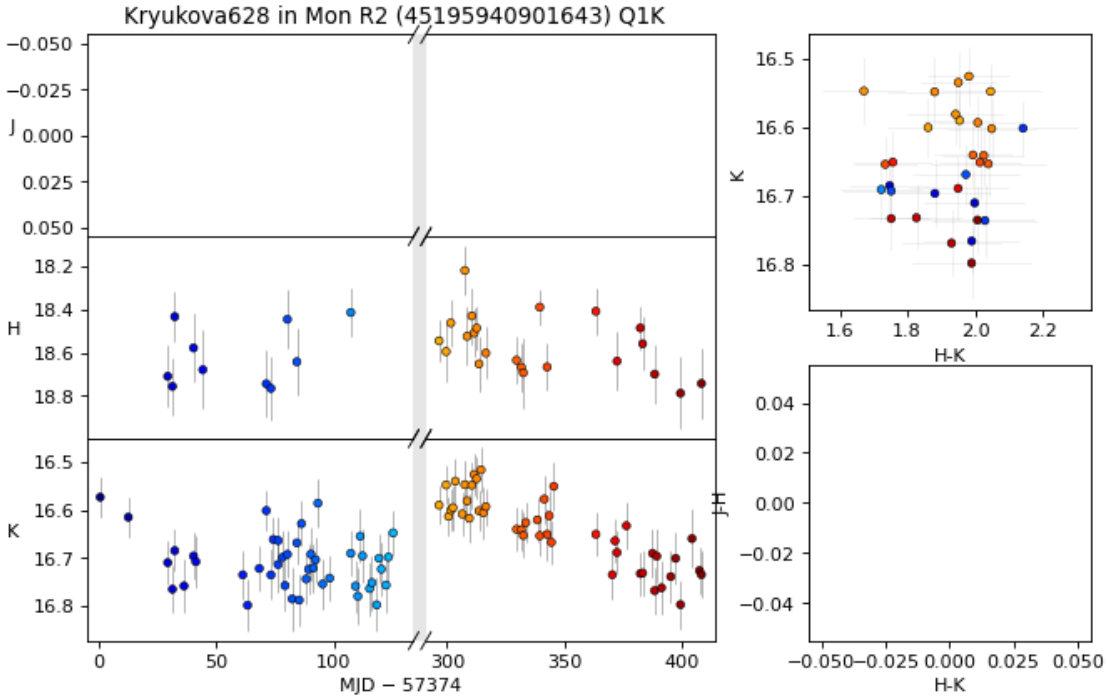
5 45195940910989



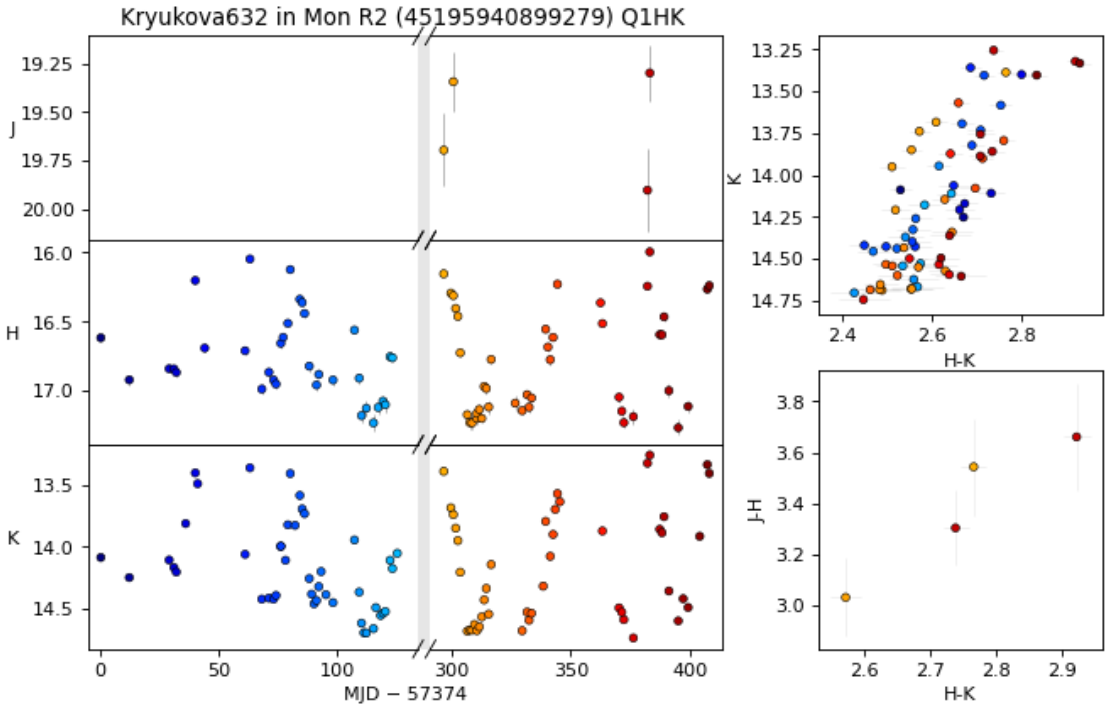
6 45195940910935



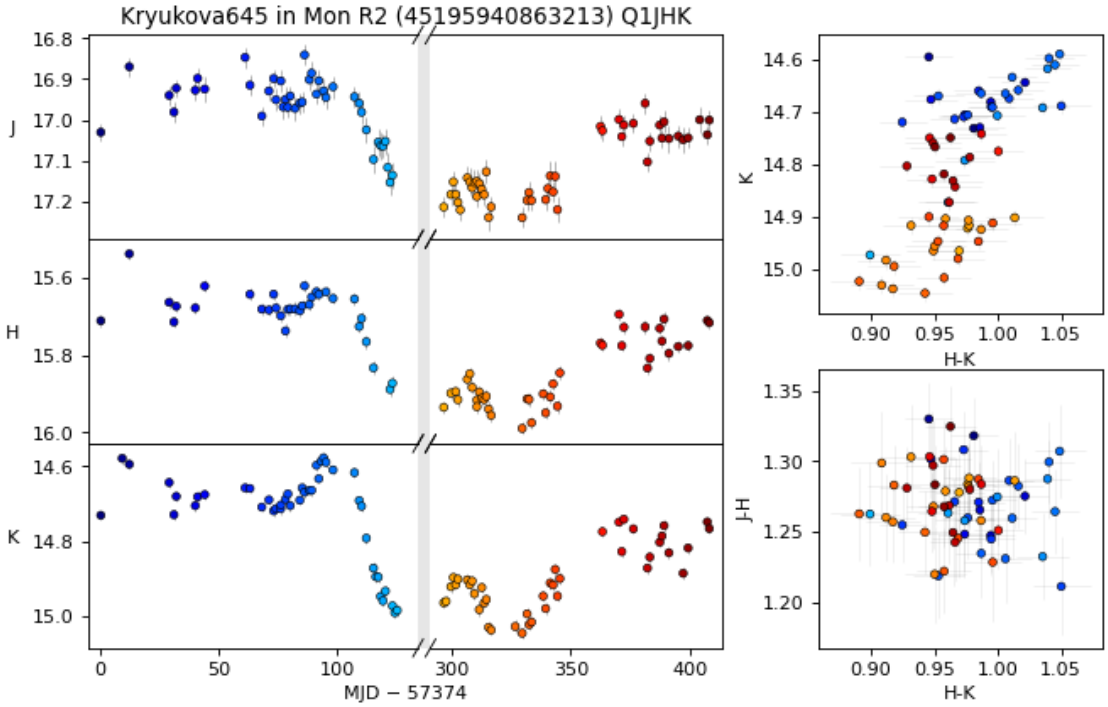
7 45195940901643



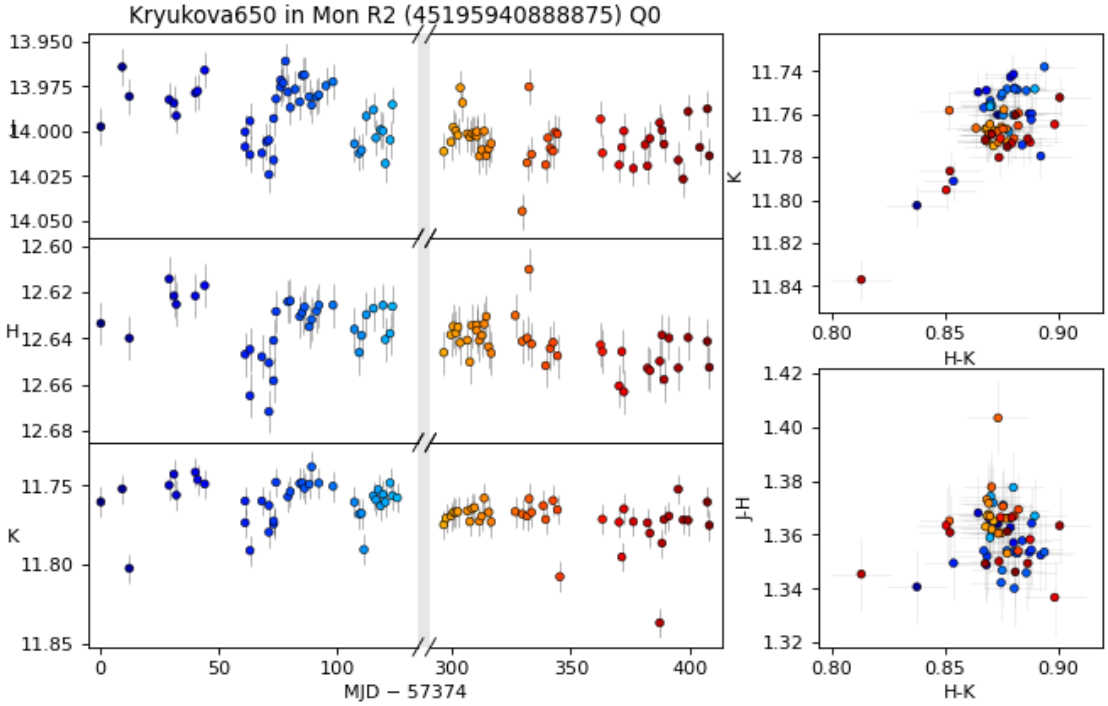
8 45195940899279



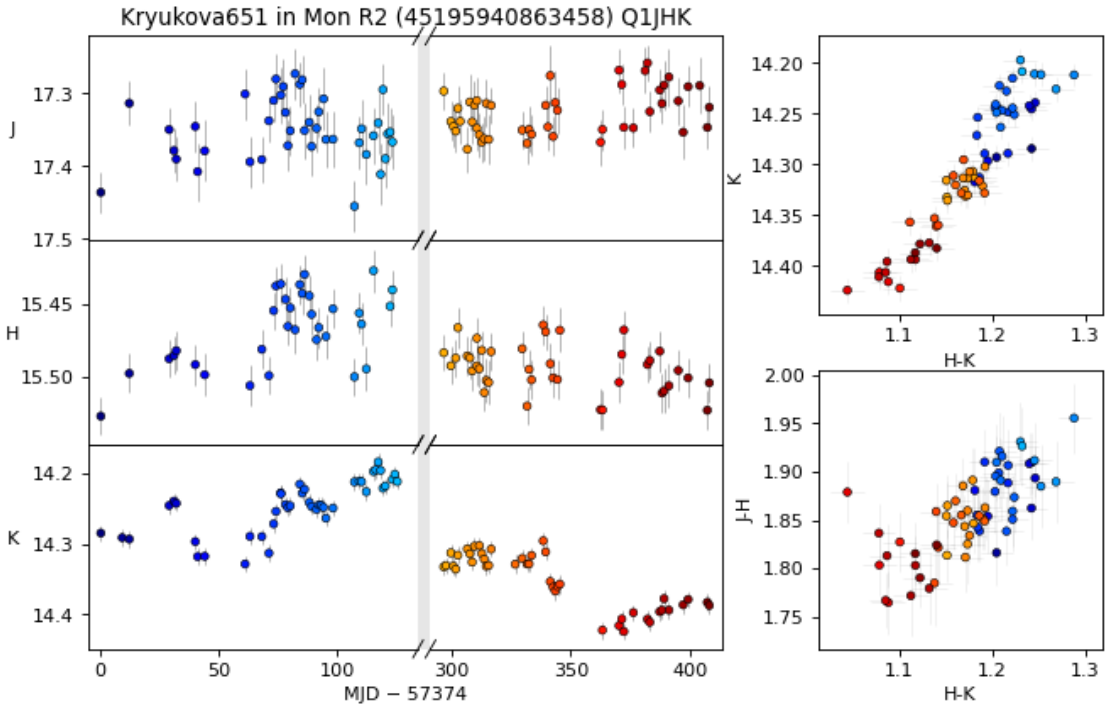
9 45195940863213



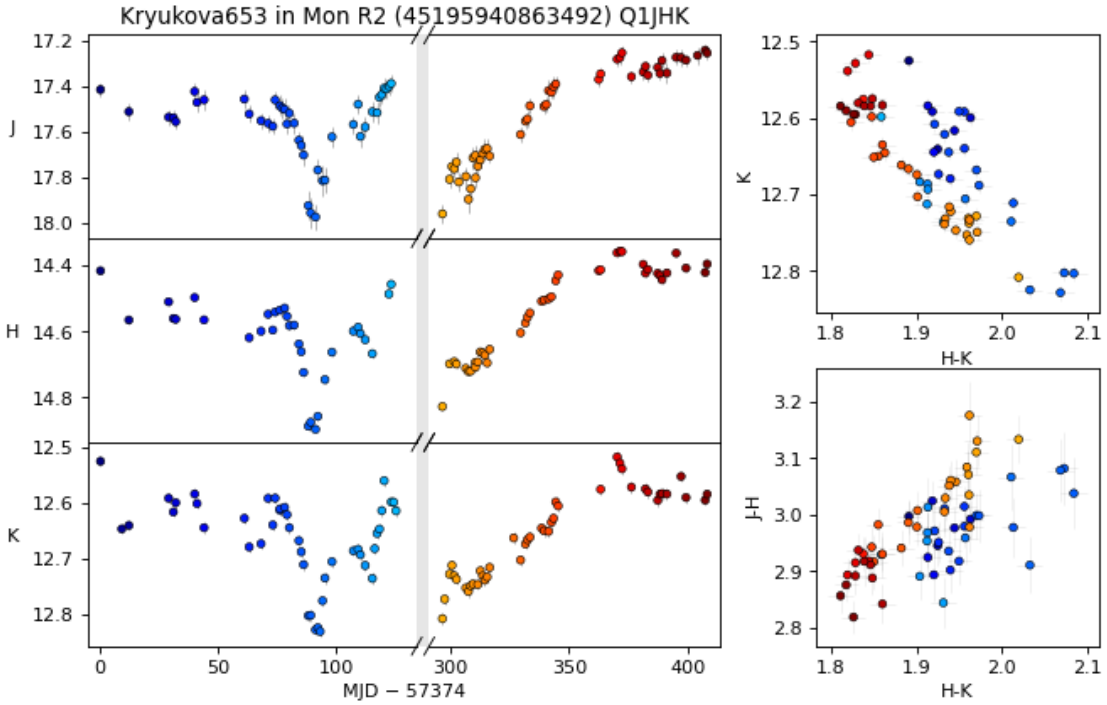
10 45195940888875



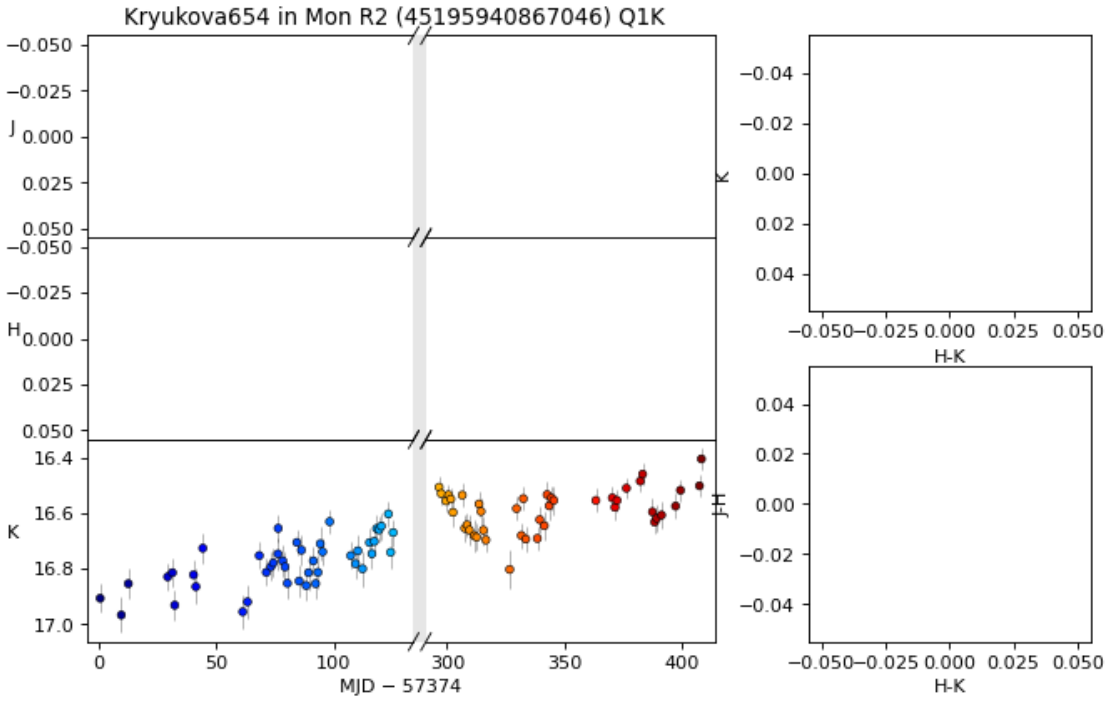
11 45195940863458



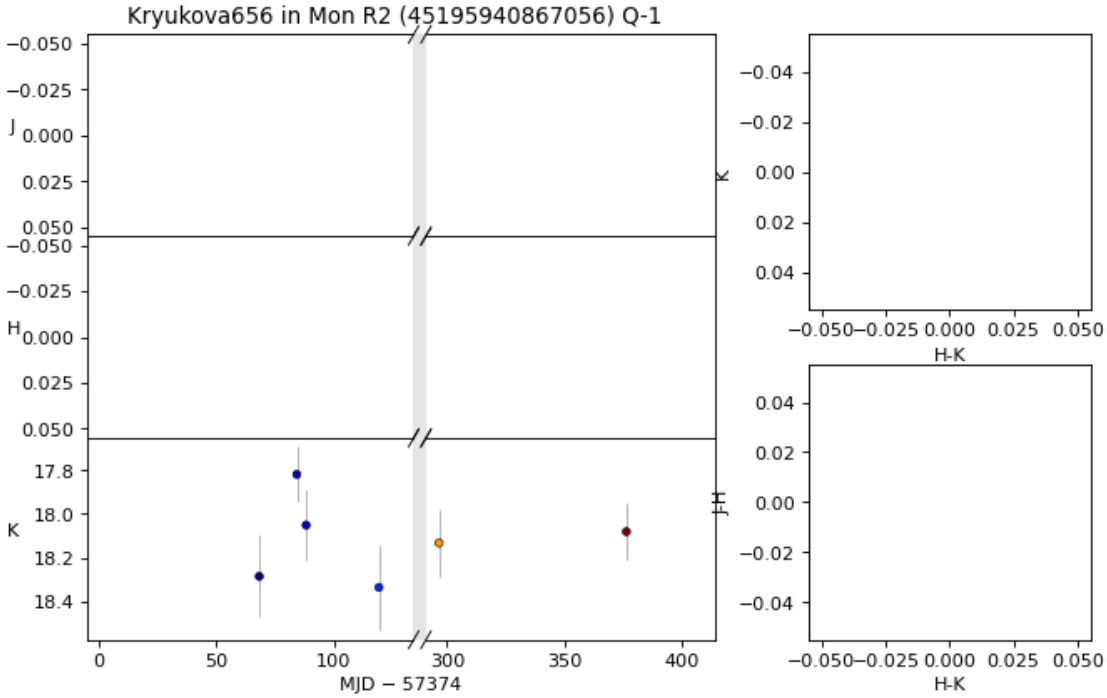
12 45195940863492



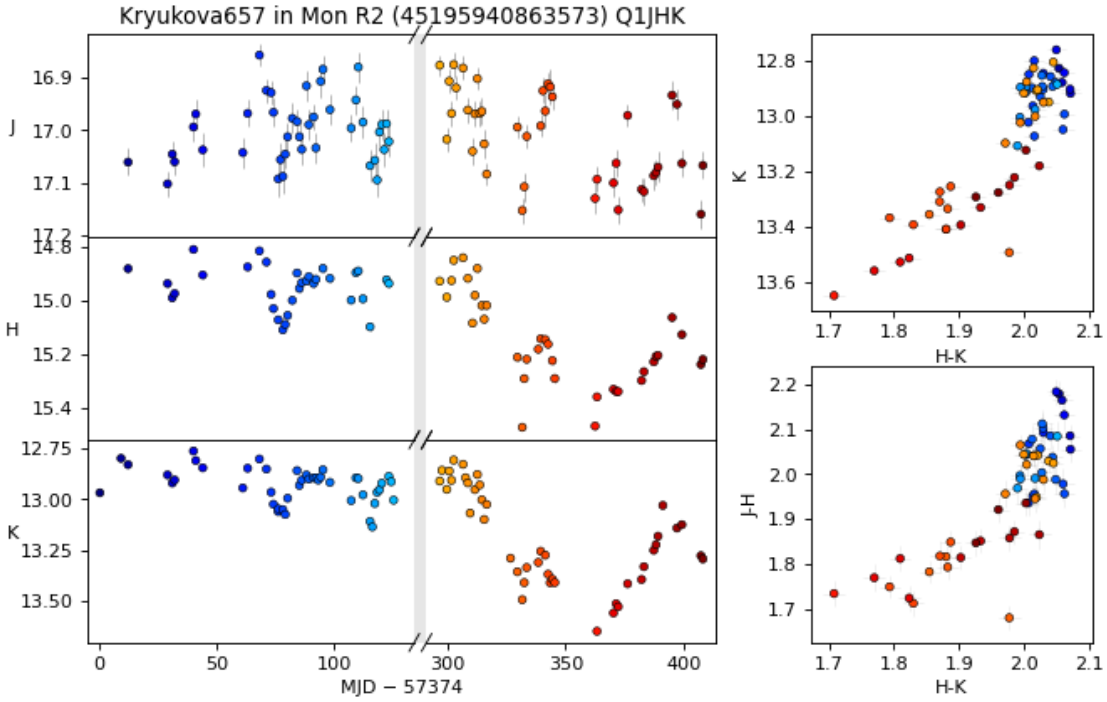
13 45195940867046



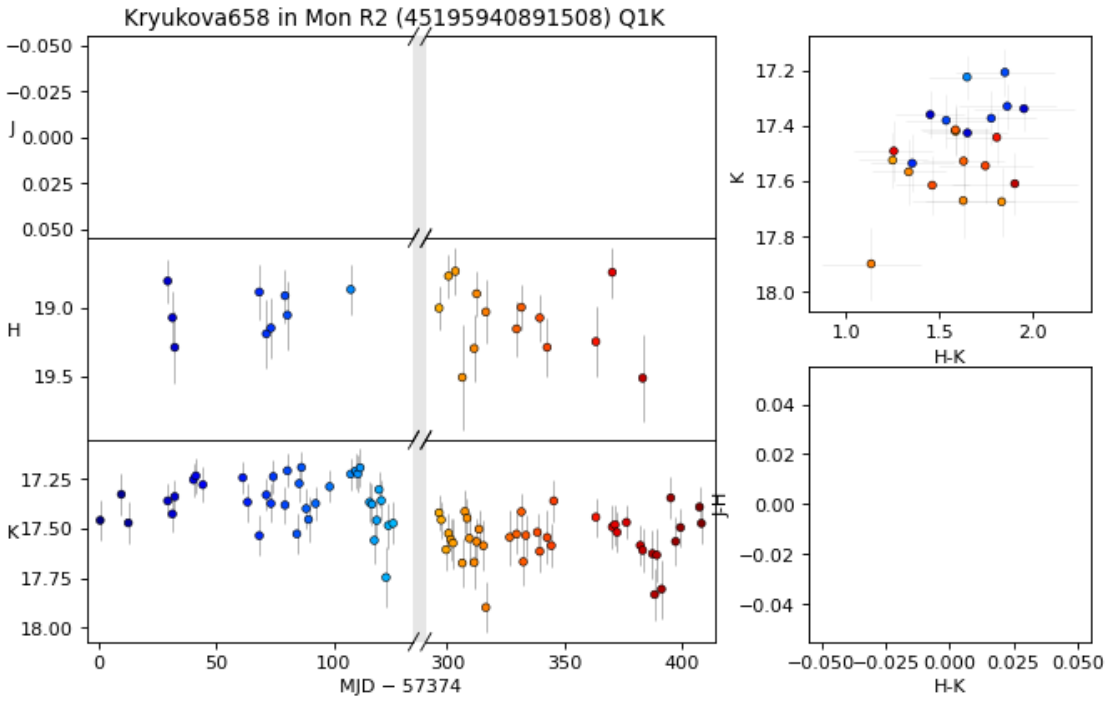
14 45195940867056



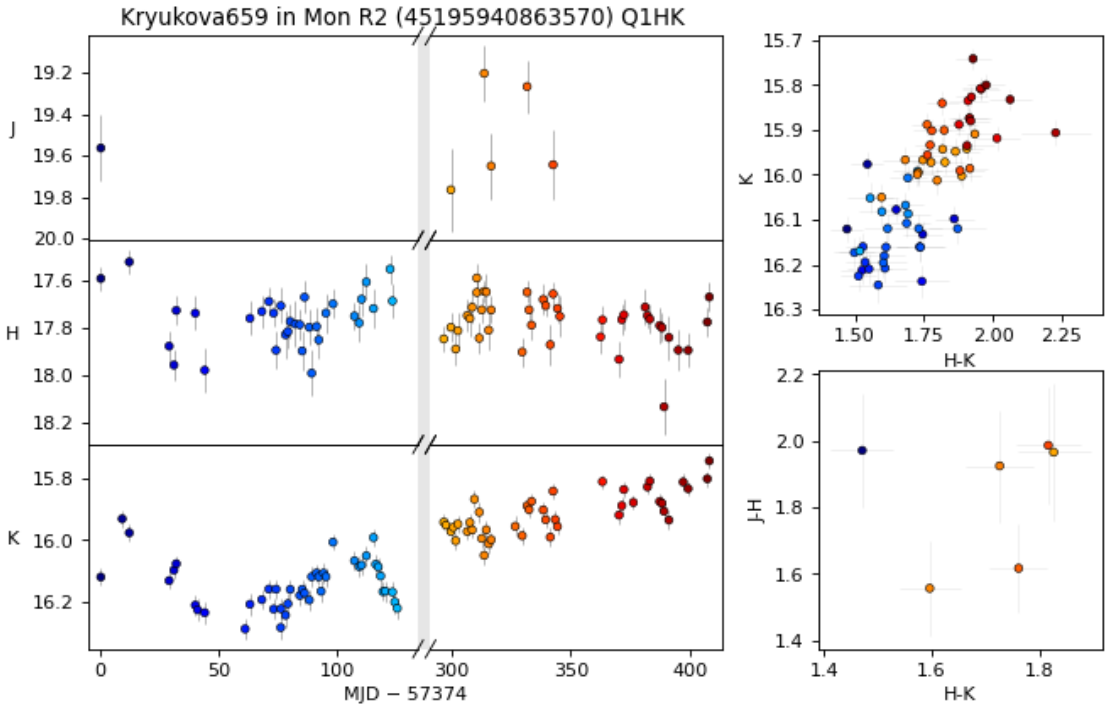
15 45195940863573



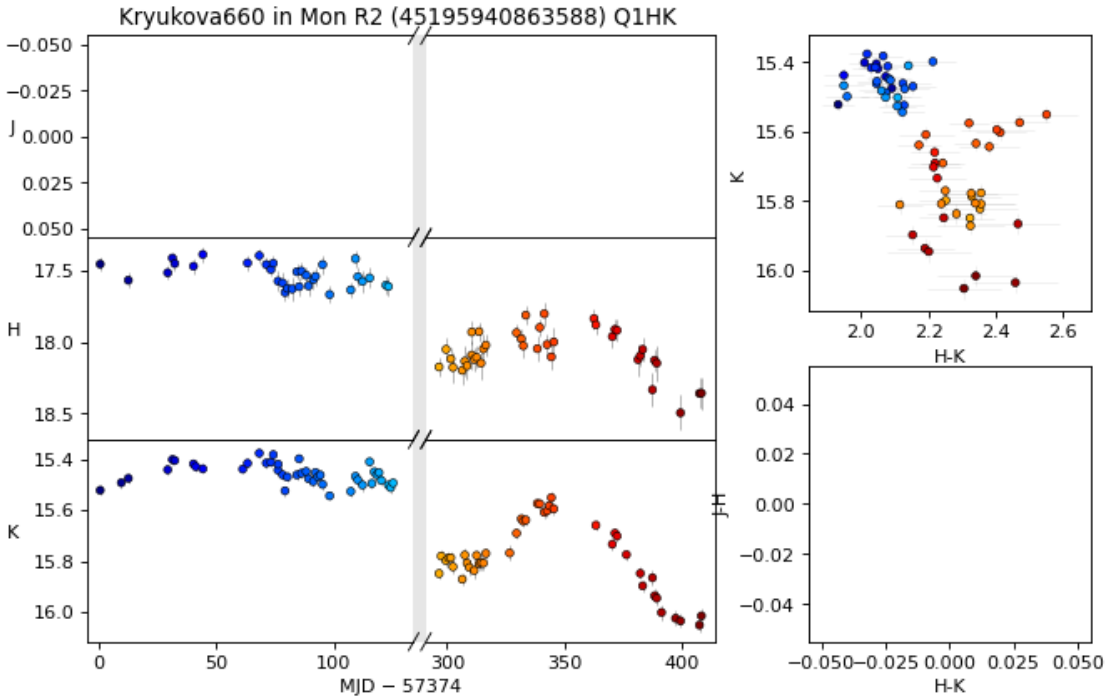
16 45195940891508



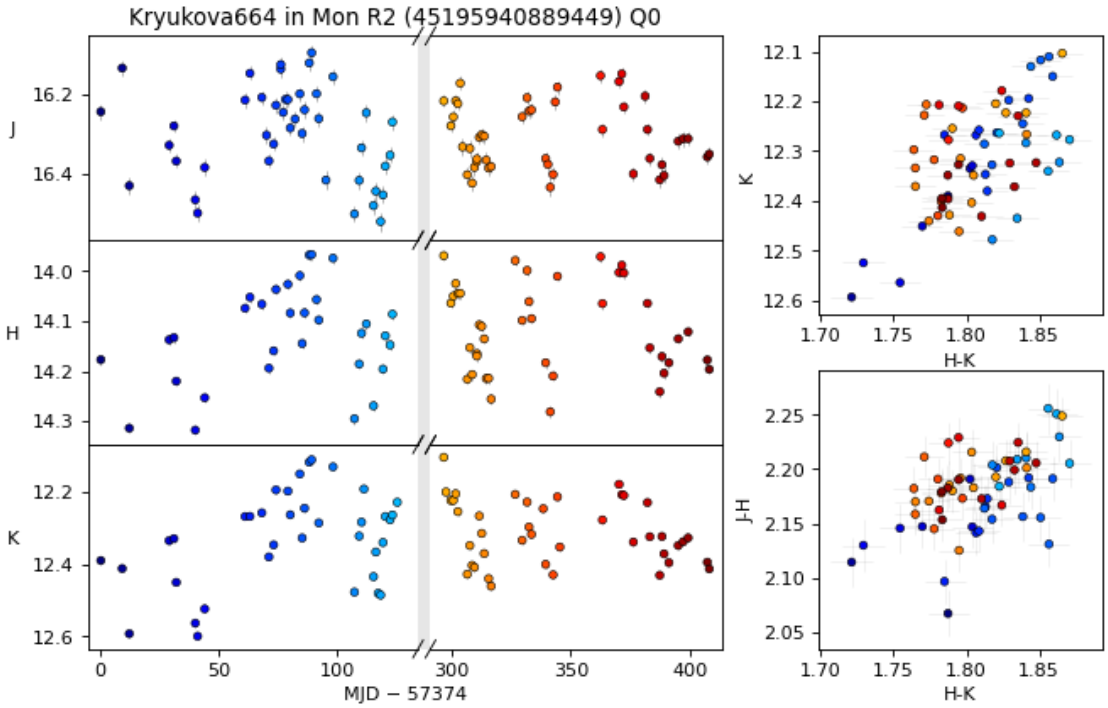
17 45195940863570



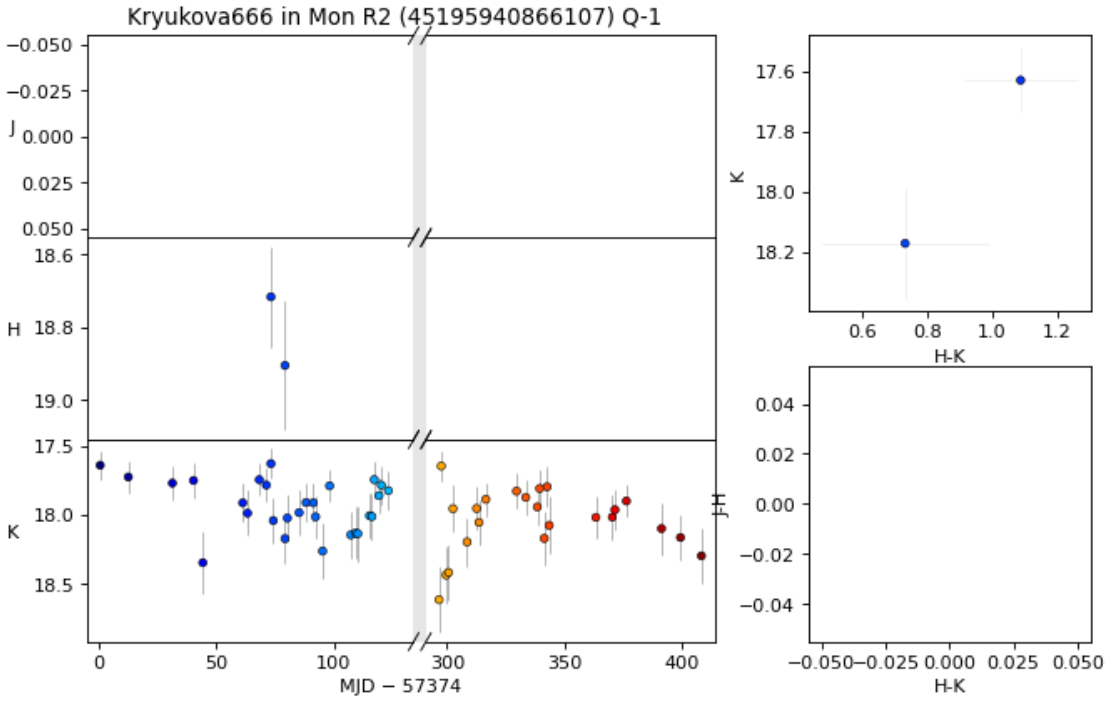
18 45195940863588



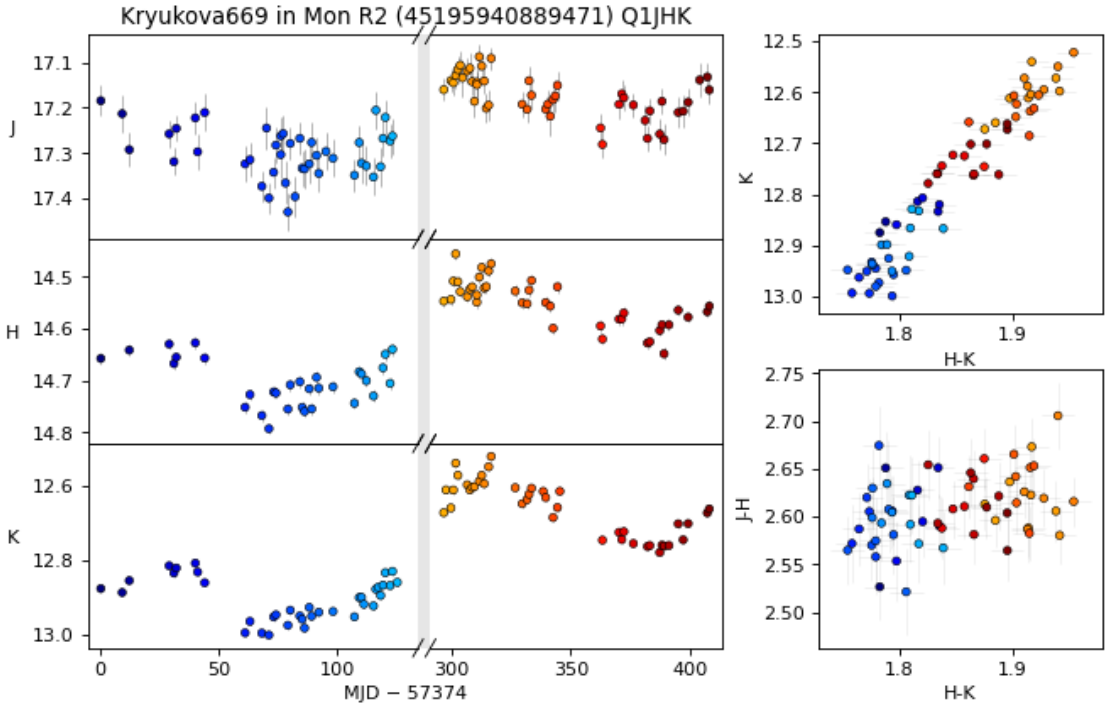
19 45195940889449



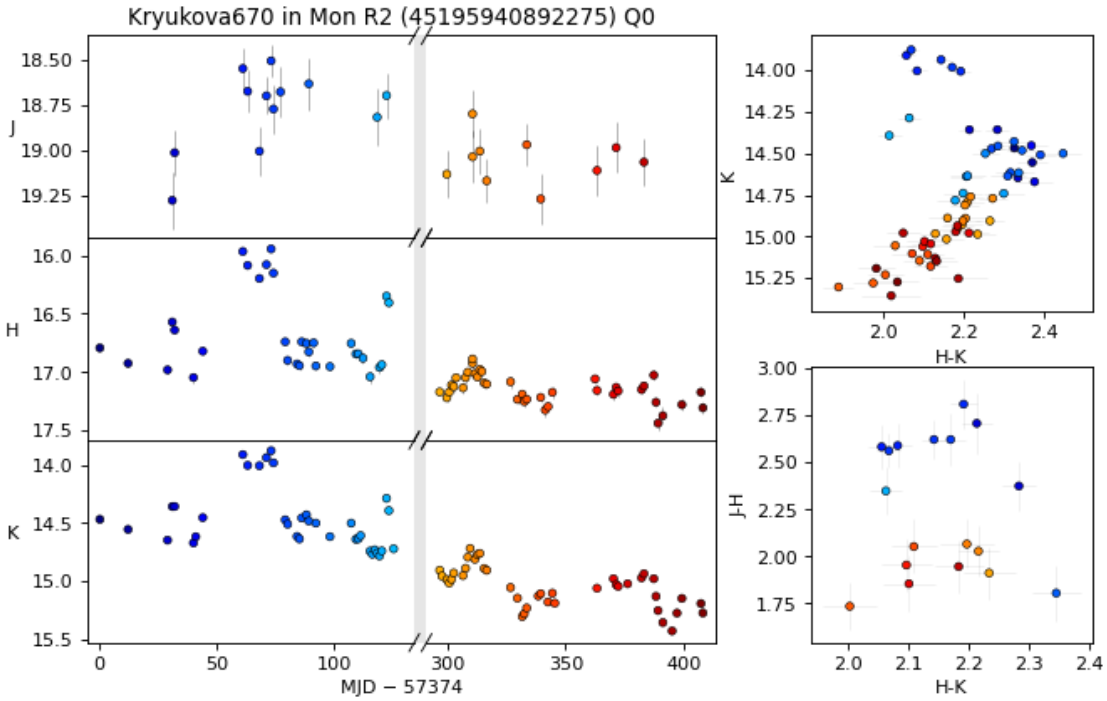
20 45195940866107



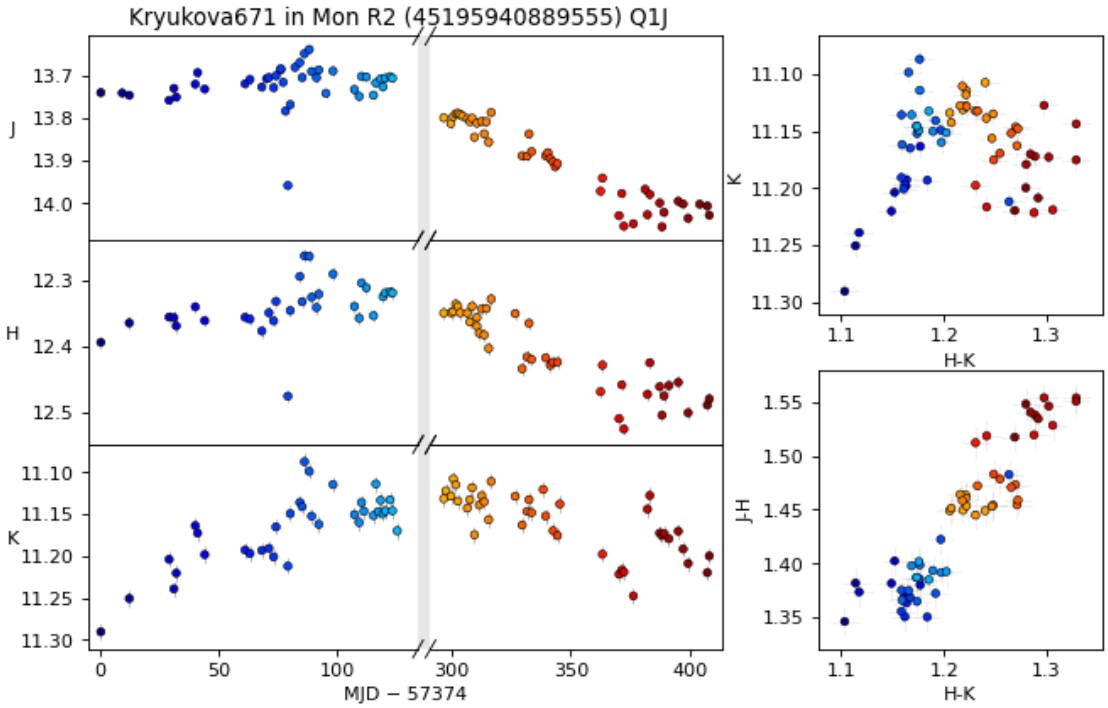
21 45195940889471



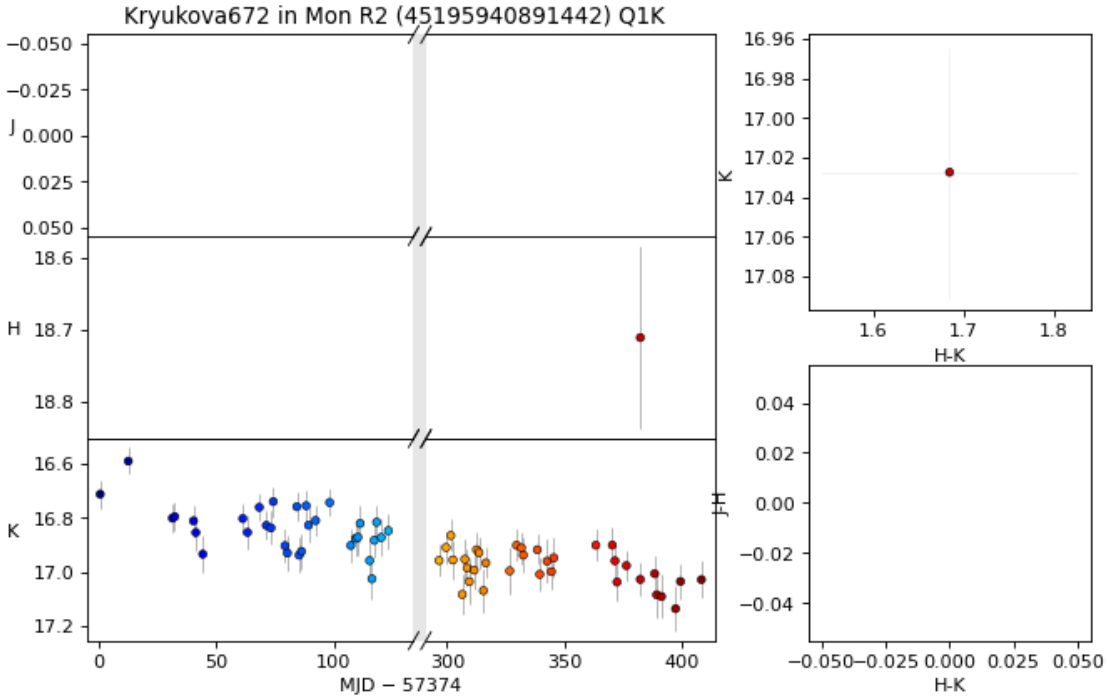
22 45195940892275



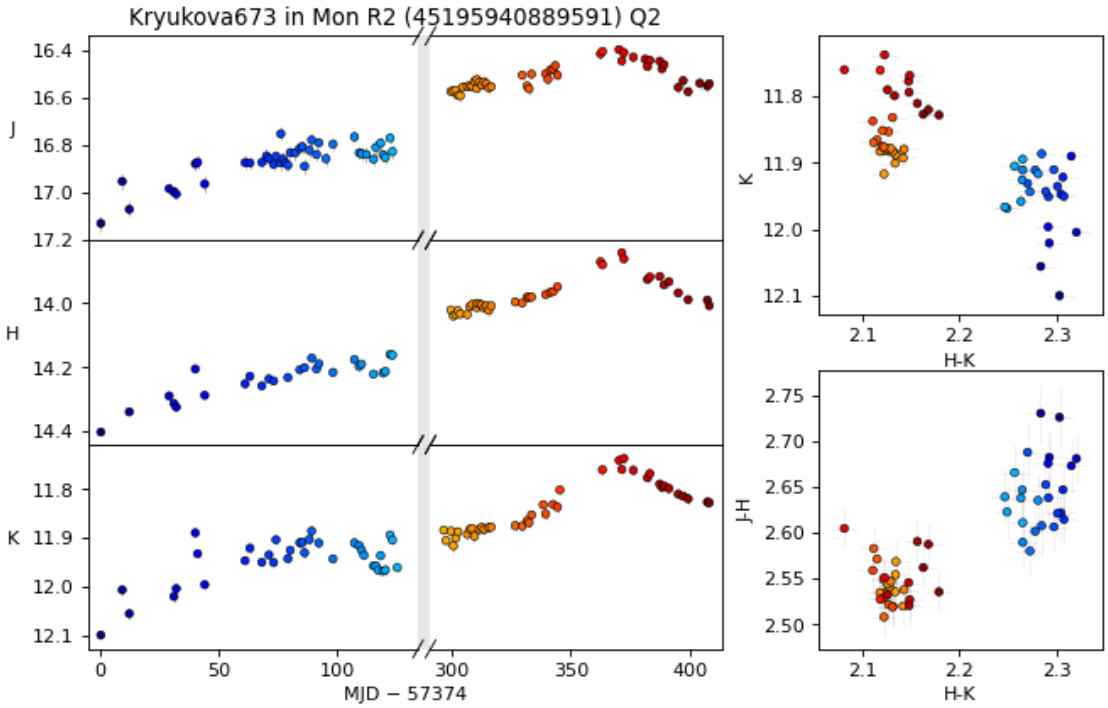
23 45195940889555



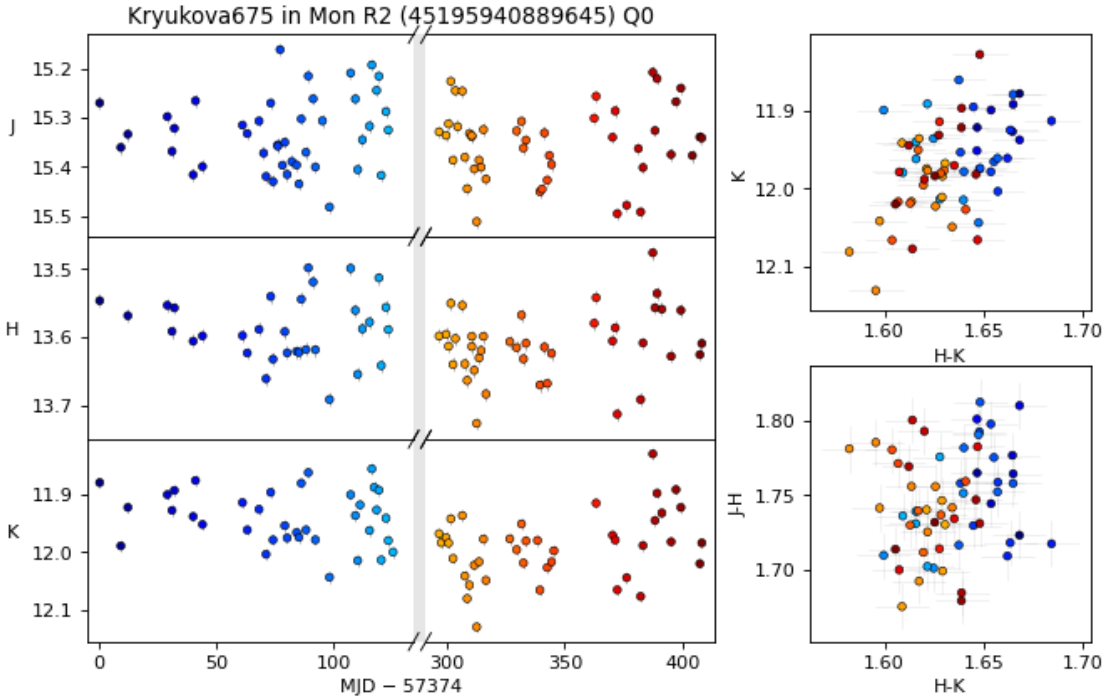
24 45195940891442



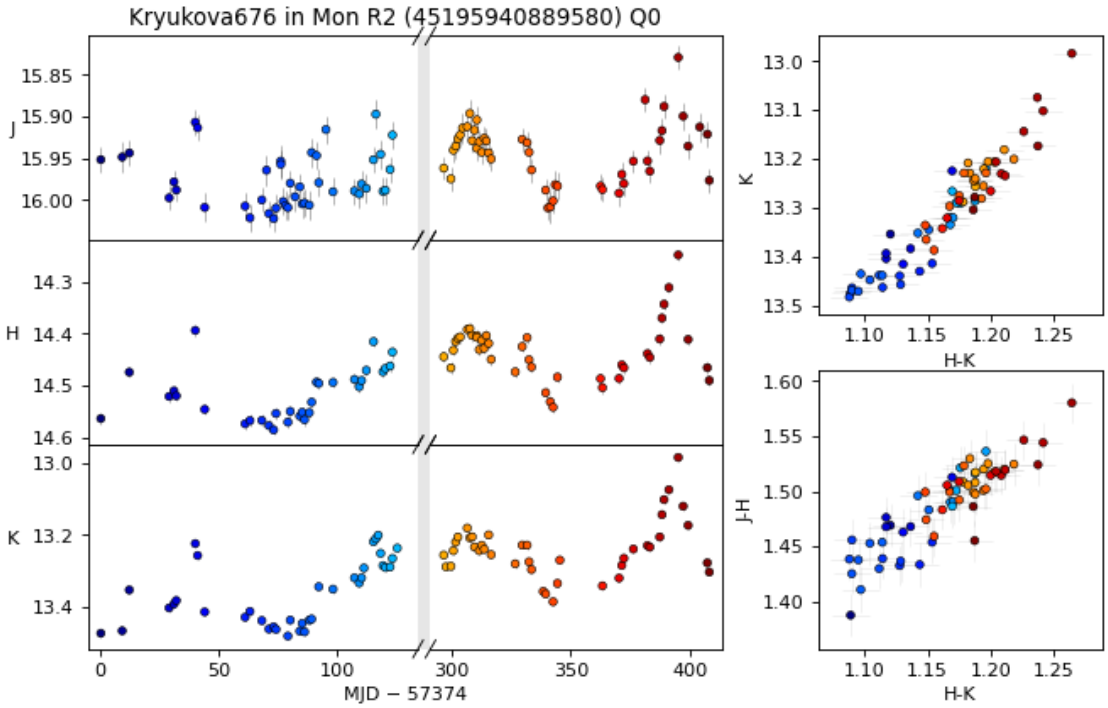
25 45195940889591



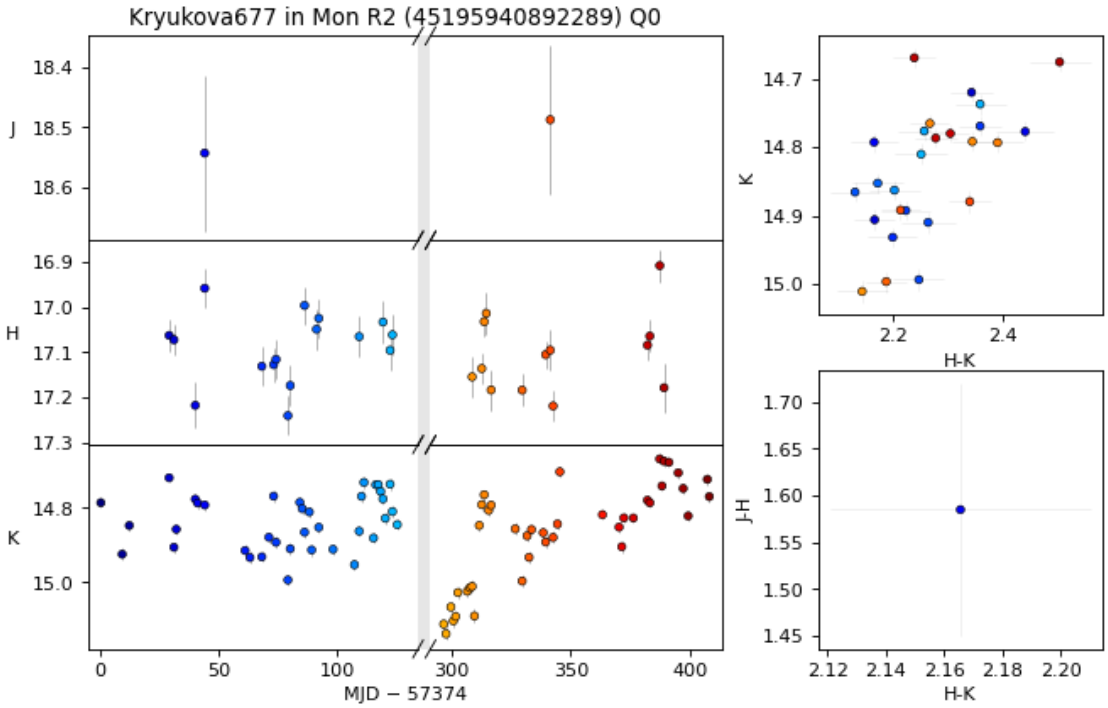
26 45195940889645



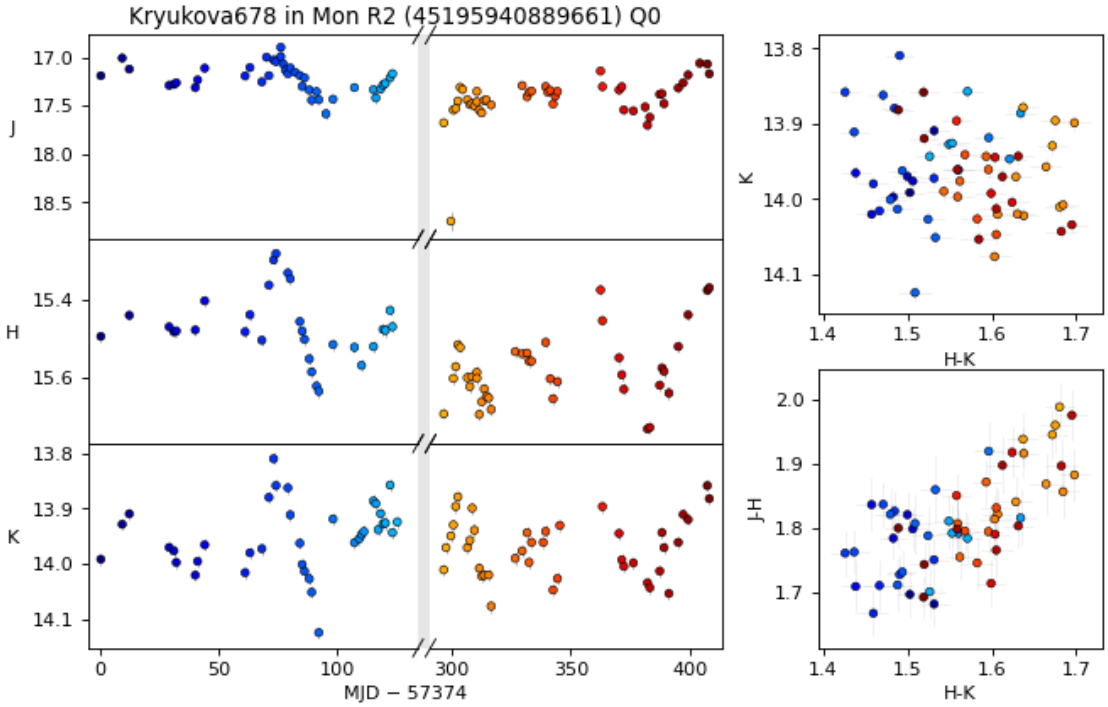
27 45195940889580



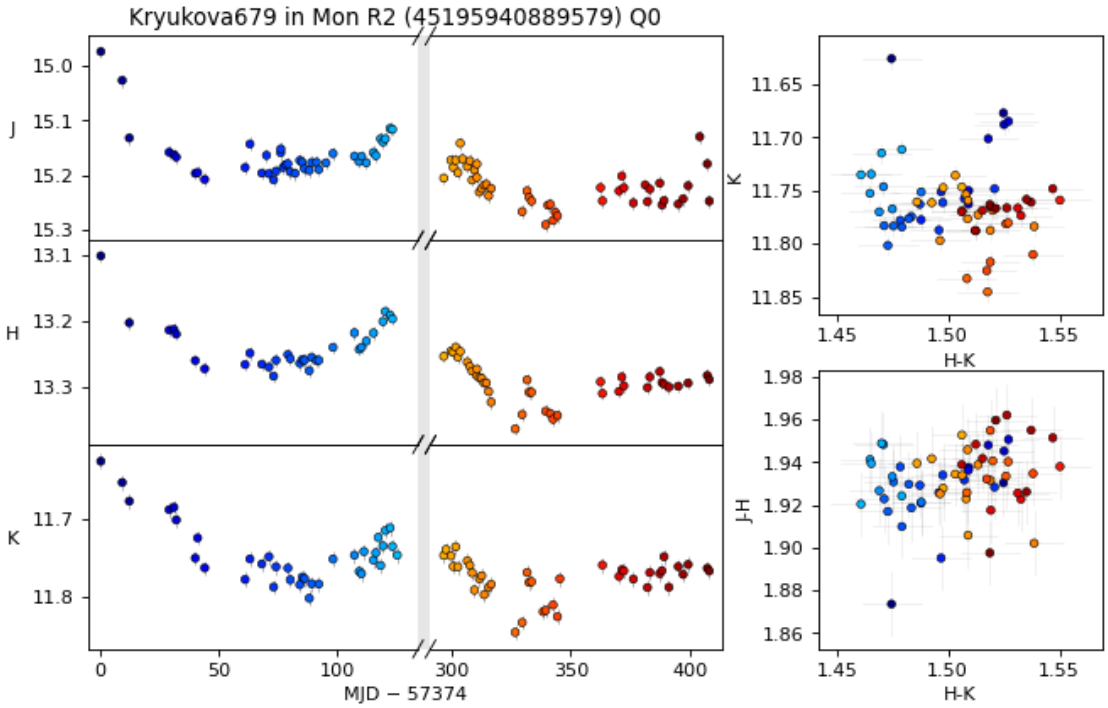
28 45195940892289



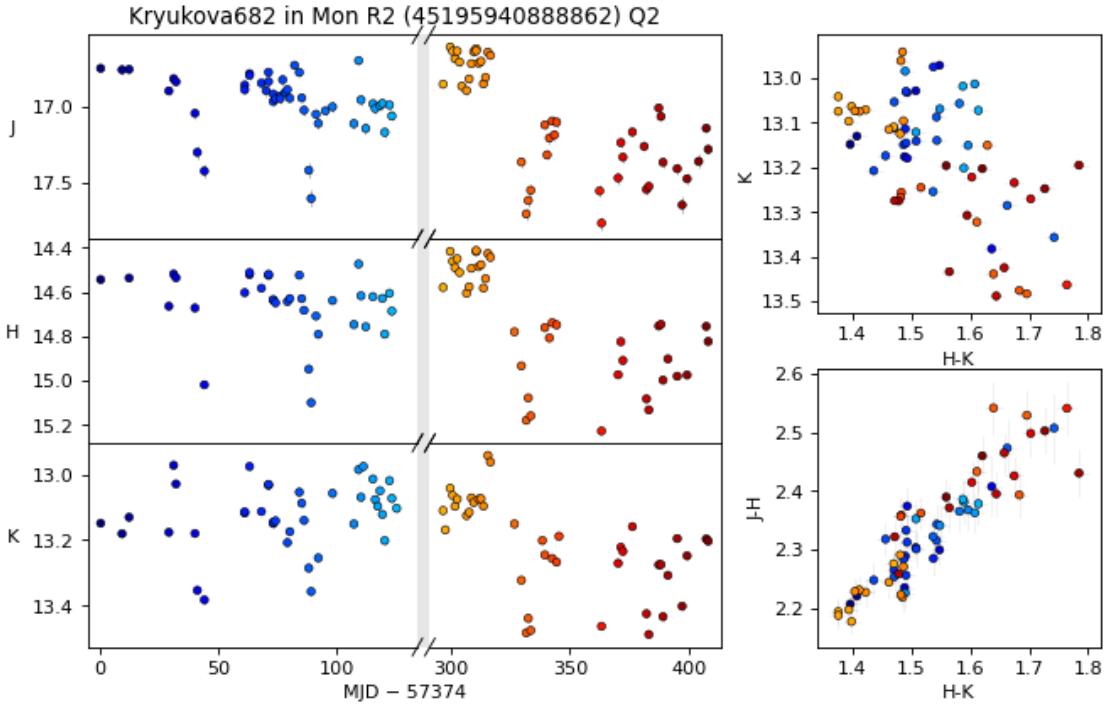
29 45195940889661



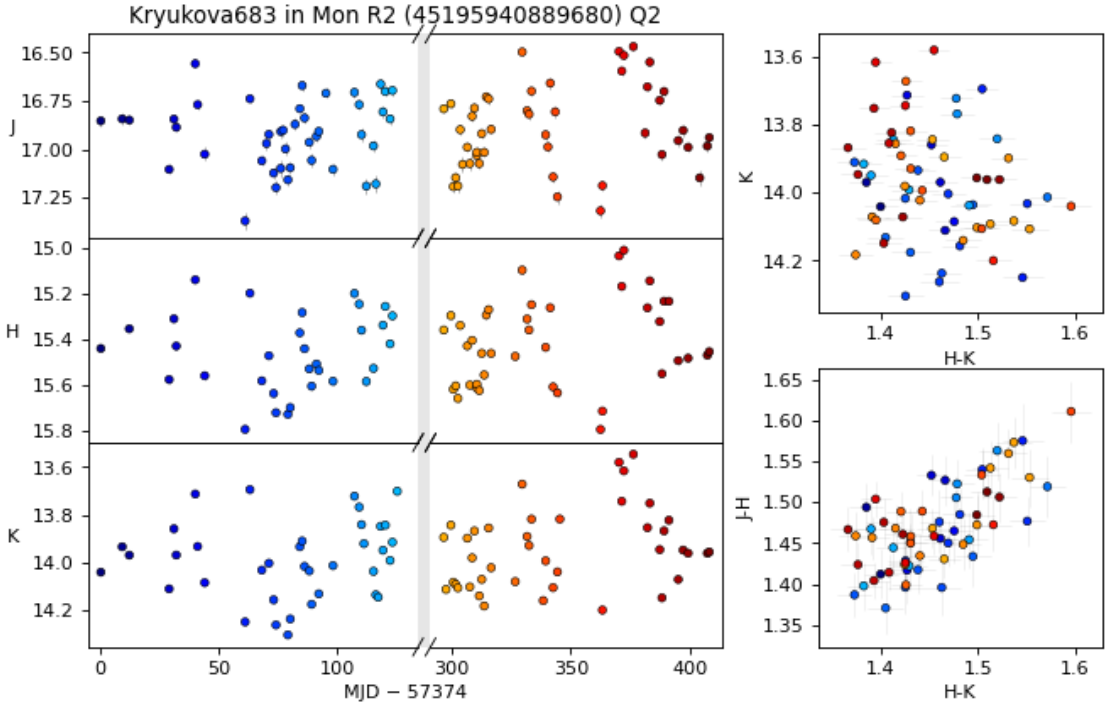
30 45195940889579



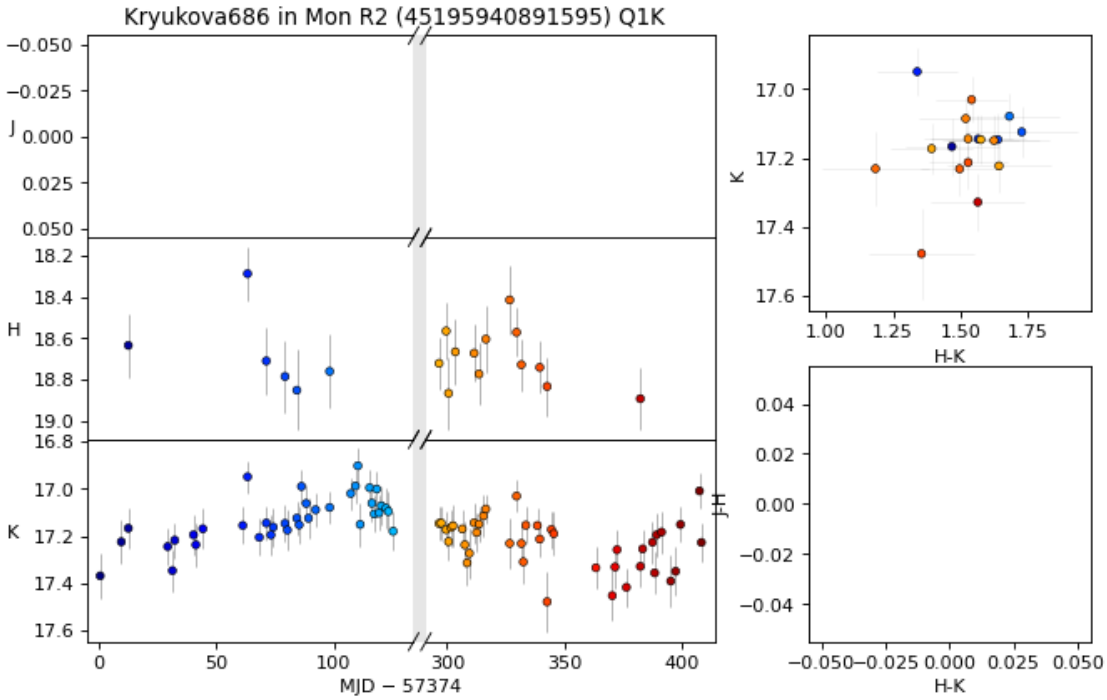
31 45195940888862



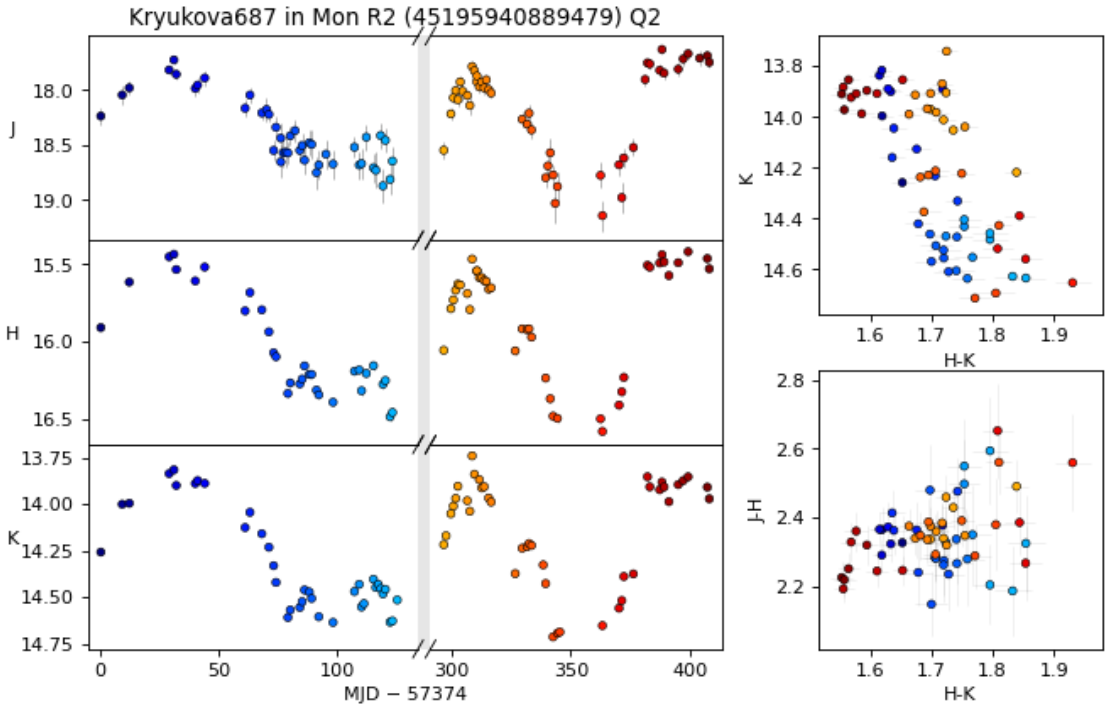
32 45195940889680



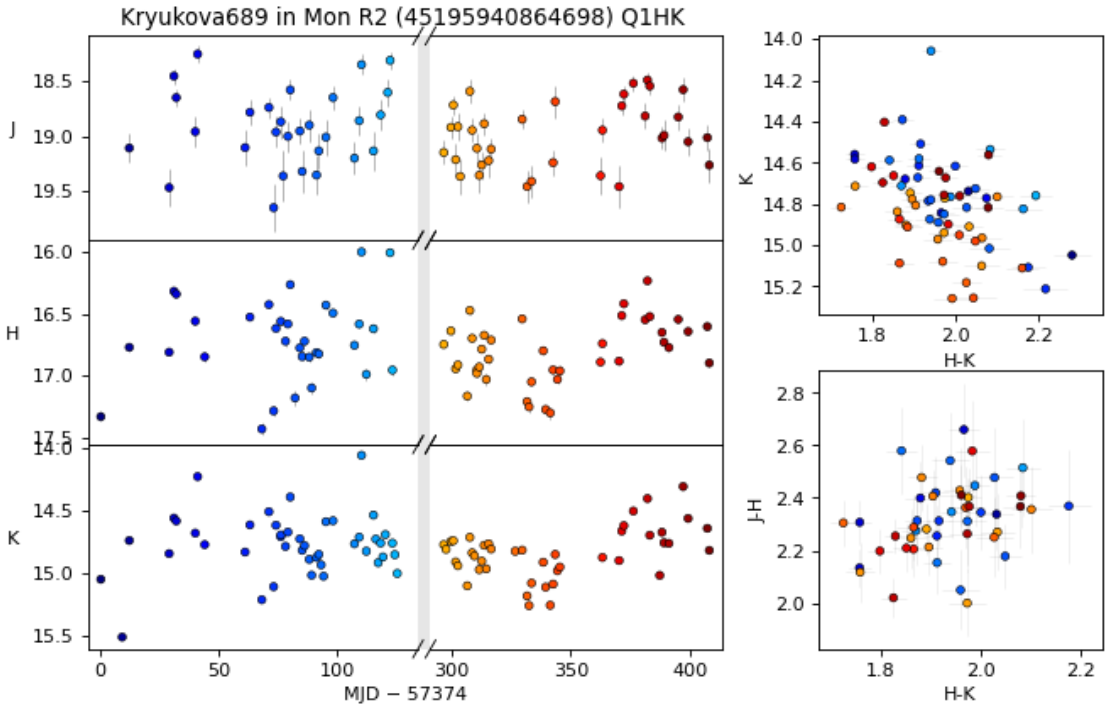
33 45195940891595



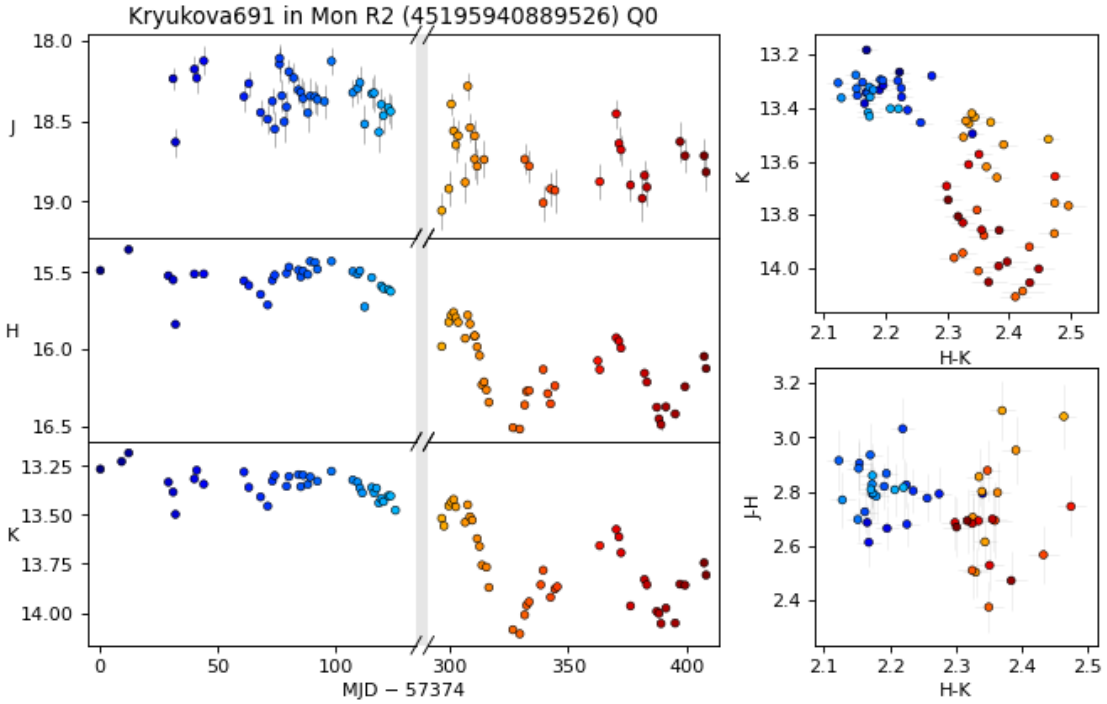
34 45195940889479



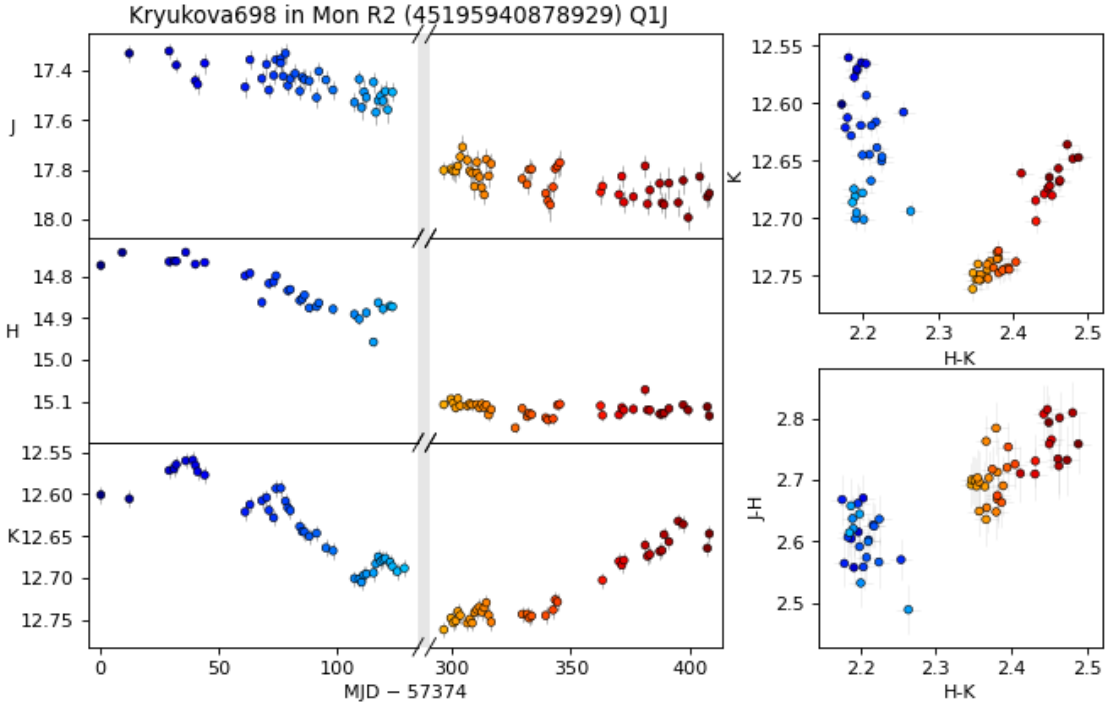
35 45195940864698



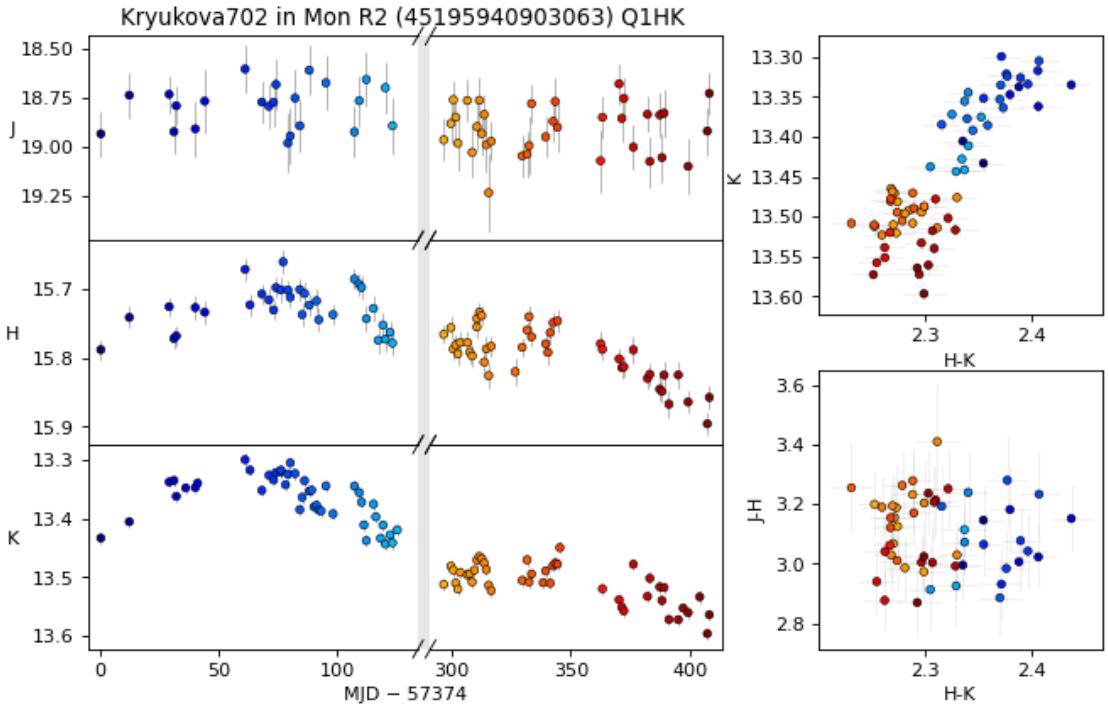
36 45195940889526



37 45195940878929



38 45195940903063



[ ]:

[ ]: *# I might want to compare with Gutermuth et al. 2009 eventually!*