Using_astroquery

April 28, 2025

1 K Using astroquery

- Astroquery: part of astropy : https://astroquery.readthedocs.io/en/latest/index.html

astroquery: An Astronomical Web-querying Package in Python Ginsburg, A., Sipocz, B. M., Brasseur, C. E., Cowperthwaite, P. S., Craig, M. W., Deil, C., Guillochon, J., Guzman, G., Liedtke, S., Lian Lim, P., Lockhart, K. E., Mommert, M., Morris, B. M., Norman, H., Parikh, M., Persson, M. V., Robitaille, T. P., Segovia, J.-C., Singer, L. P., Tollerud, E. J., de Val-Borro, M., Valtchanov, I., Woillez, J., Astroquery Collaboration, & a subset of astropy Collaboration 2019, Astronomical Journal, 157, 98

ABSTRACT: astroquery is a collection of tools for requesting data from databases hosted on remote servers with interfaces exposed on the internet, including those with web pages but without formal application program interfaces. These tools are built on the Python requests package, which is used to make HTTP requests, and astropy, which provides most of the data parsing functionality. astroquery modules generally attempt to replicate the web page interface provided by a given service as closely as possible, making the transition from browser-based to command-line interaction easy. astroquery has received significant contributions from throughout the astronomical community, including several from telescope archives. astroquery enables the creation of fully reproducible workflows from data acquisition through publication. This paper describes the philosophy, basic structure, and development model of the astroquery package. The complete documentation for astroquery can be found at http://astroquery.readthedocs.io/. ADS URL : https://ui.adsabs.harvard.edu/abs/2019AJ....157...98G

- Some examples of Astroquery uses:
- part of astroquery: TAP/TAP+: https://astroquery.readthedocs.io/en/latest/utils/tap.html
- Gaia TAP+: https://astroquery.readthedocs.io/en/latest/gaia/gaia.html
- Atomic line list: https://astroquery.readthedocs.io/en/latest/atomic/atomic.html
- Simbad: https://astroquery.readthedocs.io/en/latest/simbad/simbad.html
- Vizier: https://astroquery.readthedocs.io/en/latest/vizier/vizier.html
- This lecture explains step by step how to extract information from the Gaia database: https://allendowney.github.io/AstronomicalData/
- Talk on astroquery from ADASS XXX conference : https://www.youtube.com/watch?v=FcLvhgSHLl0

[1]:
```
import matplotlib.pyplot as plt
```

1.0.1 Querying Vizier

[2]:

```
try:
    from astroquery.vizier import Vizier
except:
    !pip install astroquery
    from astroquery.vizier import Vizier
```

```
Collecting astroquery
  Downloading astroquery-0.4.10-py3-none-any.whl.metadata (6.3 kB)
Requirement already satisfied: numpy>=1.20 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astroquery) (2.2.4)
Requirement already satisfied: astropy>=5.0 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astroquery) (7.0.0)
Requirement already satisfied: requests>=2.19 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astroquery) (2.32.3)
Requirement already satisfied: beautifulsoup4>=4.8 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astroquery) (4.12.3)
Collecting html5lib>=0.999 (from astroquery)
  Downloading html5lib-1.1-py2.py3-none-any.whl.metadata (16 kB)
Collecting keyring>=15.0 (from astroquery)
  Downloading keyring-25.6.0-py3-none-any.whl.metadata (20 kB)
Collecting pyvo>=1.5 (from astroquery)
  Downloading pyvo-1.6.2-py3-none-any.whl.metadata (4.7 kB)
Requirement already satisfied: pyerfa>=2.0.1.1 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astropy>=5.0->astroquery) (2.0.1.5)
Requirement already satisfied: astropy-iers-data>=0.2024.10.28.0.34.7 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astropy>=5.0->astroquery) (0.2025.1.13.0.34.51)
Requirement already satisfied: PyYAML>=6.0.0 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astropy>=5.0->astroquery) (6.0.2)
Requirement already satisfied: packaging>=22.0.0 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from astropy>=5.0->astroquery) (24.2)
Requirement already satisfied: soupsieve>1.2 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from beautifulsoup4>=4.8->astroquery) (2.5)
Requirement already satisfied: six>=1.9 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from html5lib>=0.999->astroquery) (1.17.0)
Requirement already satisfied: webencodings in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
```

```
packages (from html5lib>=0.999->astroquery) (0.5.1)
Collecting jaraco.classes (from keyring>=15.0->astroquery)
  Downloading jaraco.classes-3.4.0-py3-none-any.whl.metadata (2.6 kB)
Collecting jaraco.functools (from keyring>=15.0->astroquery)
  Downloading jaraco.functools-4.1.0-py3-none-any.whl.metadata (2.9 kB)
Collecting jaraco.context (from keyring>=15.0->astroquery)
  Downloading jaraco.context-6.0.1-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from requests>=2.19->astroquery) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from requests>=2.19->astroquery) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from requests>=2.19->astroquery) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/Users/christophemorisset/anaconda3/envs/Python_Lecture/lib/python3.13/site-
packages (from requests>=2.19->astroquery) (2025.1.31)
Collecting more-itertools (from jaraco.classes->keyring>=15.0->astroquery)
  Downloading more_itertools-10.7.0-py3-none-any.whl.metadata (37 kB)
Downloading astroquery-0.4.10-py3-none-any.whl (11.1 MB)
                          11.1/11.1 MB
33.2 MB/s eta 0:00:0000:010:01
Downloading html5lib-1.1-py2.py3-none-any.whl (112 kB)
Downloading keyring-25.6.0-py3-none-any.whl (39 kB)
Downloading pyvo-1.6.2-py3-none-any.whl (999 kB)
                          999.4/999.4 kB
46.2 MB/s eta 0:00:00
Downloading jaraco.classes-3.4.0-py3-none-any.whl (6.8 kB)
Downloading jaraco.context-6.0.1-py3-none-any.whl (6.8 kB)
Downloading jaraco.functools-4.1.0-py3-none-any.whl (10 kB)
Downloading more_itertools-10.7.0-py3-none-any.whl (65 kB)
Installing collected packages: more-itertools, jaraco.context, html5lib,
jaraco.functools, jaraco.classes, pyvo, keyring, astroquery
Successfully installed astroquery-0.4.10 html5lib-1.1 jaraco.classes-3.4.0
jaraco.context-6.0.1 jaraco.functools-4.1.0 keyring-25.6.0 more-itertools-10.7.0
pyvo-1.6.2
```

[3]:

```
catalog_list = Vizier.find_catalogs('Kang W51')
```

[4]:

```
catalog_list
```

[4]:

```
OrderedDict([('J/ApJ/684/1143', </>),
             ('J/ApJ/736/87', </>),
             ('J/ApJ/738/79', </>),
             ('J/ApJ/760/12', </>),
('J/ApJ/785/119', </>),
('J/ApJ/813/39', </>),
('J/ApJ/839/12', </>),
('J/ApJ/859/4', </>),
('J/ApJ/879/10', </>),
('J/ApJ/886/75', </>),
('J/ApJ/886/93', </>),
('J/ApJ/887/134', </>),
('J/ApJ/889/L34', </>),
('J/ApJ/892/93', </>),
('J/ApJ/902/104', </>),
('J/ApJ/916/47', </>),
('J/ApJ/920/L45', </>),
('J/ApJS/165/360', </>),
('J/ApJS/191/232', </>),
('J/ApJS/236/51', </>),
('J/ApJS/238/29', </>),
('J/ApJS/249/33', </>),
('J/ApJS/258/40', </>),
('J/ApJS/265/4', </>),
('J/ApJS/269/25', </>),
('J/ApJS/271/25', </>),
('J/ApJS/272/22', </>),
('J/A+A/454/717', </>),
('J/A+A/548/A29', </>),
('J/A+A/563/A120', </>),
('J/A+A/578/A51', </>),
('J/A+A/622/A81', </>),
('J/A+A/642/A85', </>),
('J/A+A/647/A78', </>),
('J/A+A/651/A74', </>),
('J/AJ/127/539', </>),
('J/AJ/128/846', </>),
('J/AJ/144/35', </>),
('J/AJ/144/150', </>),
('J/AJ/145/167', </>),
('J/AJ/149/59', </>),
('J/AJ/150/1', </>),
('J/MNRAS/310/982', </>),
('J/MNRAS/359/865', </>),
('J/MNRAS/385/2225', </>),
('J/MNRAS/401/160', </>),
('J/MNRAS/439/611', </>),
('J/MNRAS/485/2895', </>),
('J/MNRAS/496/2790', </>),
('J/MNRAS/496/2821', </>),
('J/MNRAS/505/2801', </>)])
```

[5]:
```
catalogs = Vizier.get_catalogs('J/ApJ/706/83/ysos')
print(catalogs)
```

```
TableList with 1 tables:
        '0:J/ApJ/706/83/ysos' with 22 column(s) and 50 row(s)
```

[6]:
```
Vizier.ROW_LIMIT = -1
catalogs = Vizier.get_catalogs('J/ApJ/706/83/ysos')
print(catalogs)
```

```
TableList with 1 tables:
        '0:J/ApJ/706/83/ysos' with 22 column(s) and 737 row(s)
```

[7]:
```
Table = catalogs['J/ApJ/706/83/ysos']
```

[8]:
```
type(Table)
```

[8]:
```
astropy.table.table.Table
```

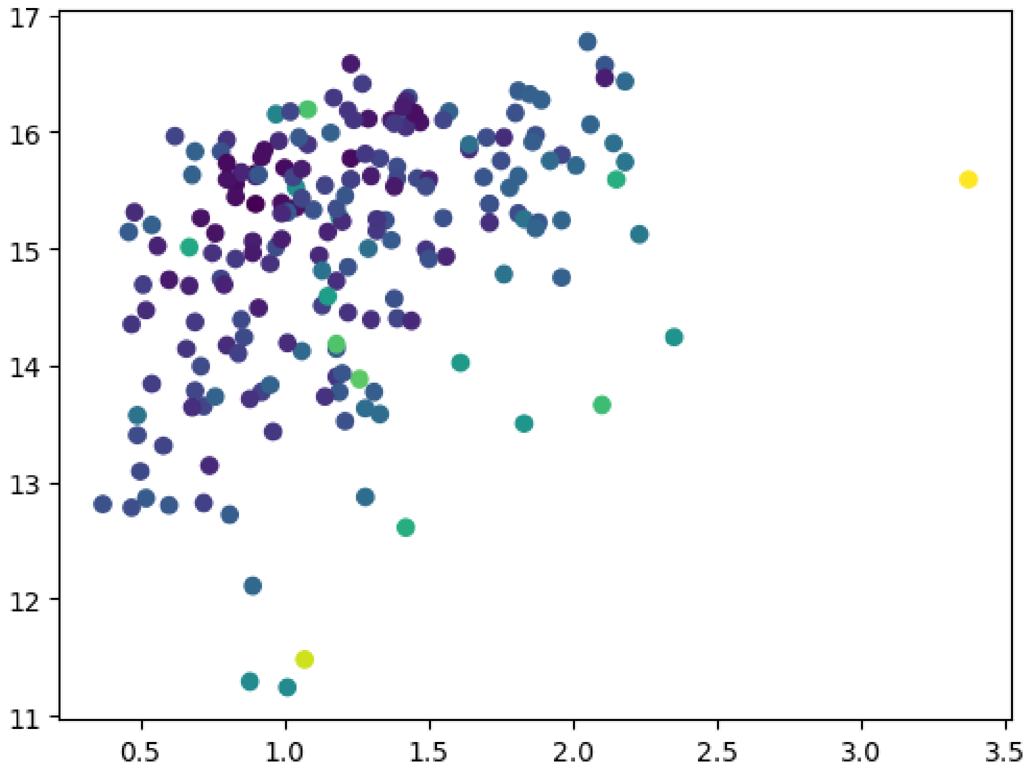
[9]:
```
print(Table)
```

```
Seq f_Seq      SSTGLMC       AV   Mstar   …  A   2M Simbad   _Glon     _Glat
                            mag  solMass  …                   deg       deg
--- ----- ----------------- ---- -------- … --- --- ------ --------- ---------
  1       G048.7567-00.6341  4.0      3.0 …   F  2M Simbad   48.7567   -0.6341
  2       G048.7579-00.2797  1.6      2.7 …   F  2M Simbad   48.7579   -0.2797
  3       G048.7605-00.0388 33.3      4.1 …   W  2M Simbad   48.7605   -0.0388
  4       G048.7618+00.0627 51.8      4.0 …   W  2M Simbad   48.7618    0.0627
  5       G048.7637+00.2022 25.7      3.5 …   W  2M Simbad   48.7637    0.2022
  6       G048.7655+00.1017  3.0      3.2 …   F  2M Simbad   48.7655    0.1017
  7       G048.7667+00.0766  7.5      4.0 …   F  2M Simbad   48.7667    0.0766
  8       G048.7702-00.1505 18.8      3.9 …   W  2M Simbad   48.7702   -0.1505
  9       G048.7703+00.0786 52.7      6.6 …   W  2M Simbad   48.7703    0.0786
…   …               …  …     … … … …    …      …      …
728       G049.9895-00.0079 20.0      4.9 …   W  2M Simbad   49.9895   -0.0079
729       G049.9899-00.1413 21.5     10.1 …   W  2M Simbad   49.9899   -0.1413
730       G049.9914-00.1333 23.5      9.1 …   W  2M Simbad   49.9914   -0.1333
731       G049.9937+00.0231  6.4      3.2 …   F  2M Simbad   49.9937    0.0231
732       G049.9952-00.0063 12.5      4.9 …   W  2M Simbad   49.9952   -0.0063
733       G049.9972+00.2599 33.6      5.6 …   W  2M Simbad   49.9972    0.2599
734       G049.9973+00.2605 17.3     10.1 …   W  2M Simbad   49.9973    0.2605
735       G049.9973+00.2630  8.4      5.2 …   F  2M Simbad   49.9973    0.2630
736       G049.9977-00.1261 11.3      8.3 …   W  2M Simbad   49.9977   -0.1261
737       G049.9982-00.1303 25.9      8.1 …   W  2M Simbad   49.9982   -0.1303
Length = 737 rows
```

[10]:
```
print(Table.keys())
```

```
['Seq', 'f_Seq', 'SSTGLMC', 'AV', 'Mstar', 'Ltot', 'Stg', 'Cl1', 'Cl2', 'Jmag',
'Hmag', 'Ksmag', '[3.6]', '[4.5]', '[5.8]', '[8.0]', '[24]', 'A', '2M',
'Simbad', '_Glon', '_Glat']
```

```
[11]: f, ax = plt.subplots()
      ax.scatter(Table['Jmag'] - Table['Hmag'], Table['Jmag'], c=Table['Mstar']);
```



1.0.2 Querying MAST

https://archive.stsci.edu/access-mast-data

```
[12]: from astroquery.mast import Catalogs, Observations
```

```
[13]: obs_table = Observations.query_object("IC 418",radius=".02 deg")
```

```
[15]: obs_table
```

```
[15]: <Table masked=True length=353>
      intentType obs_collection provenance_name … obsid        distance
        str11        str11           str31      …  str9         float64
      ---------- -------------- --------------- … --------- --------------------
         science           TESS            SPOC … 61577646                  0.0
```

```
 science        TESS          SPOC … 62280345                0.0
 science        TESS          SPOC … 28210379                0.0
 science SPITZER_SHA  SSC Pipeline …  1746918  0.821401040670269
 science SPITZER_SHA  SSC Pipeline …  1746918  0.821401040670269
 science SPITZER_SHA  SSC Pipeline …  1746918  0.821401040670269
 science SPITZER_SHA  SSC Pipeline …  1746918   0.82916807067517
 science SPITZER_SHA  SSC Pipeline …  1746918   0.82916807067517
 science SPITZER_SHA  SSC Pipeline …  1746918   0.82916807067517
     …           …             … …        …                  …
 science         HLA           HLA … 25723603                0.0
 science         HLA           HLA … 25723604                0.0
 science         HLA           HLA … 26169487                0.0
 science         HLA           HLA … 26169488                0.0
 science         HLA           HLA … 25723605                0.0
 science         HLA           HLA … 25723606                0.0
 science         HLA           HLA … 25723607                0.0
 science         HLA           HLA … 25723608                0.0
 science        FUSE            -- …   345403                0.0
 science        BEFS            -- …   347967 0.0871622250930436
```

```
[16]: mask_spectrum = obs_table['dataproduct_type'] == 'spectrum'
```

```
[17]: print(len(obs_table), mask_spectrum.sum())
```

```
353 167
```

```
[18]: data_products_by_obs = Observations.get_product_list(obs_table[mask_spectrum][0:
        ↪2])
```

```
[20]: data_products_by_obs
```

```
[20]: <Table masked=True length=154>
       obsID  obs_collection dataproduct_type … dataRights calib_level filters
        str7      str11            str8       …    str6       int64      str6
      ------- -------------- ---------------- … ---------- ----------- -------
      1746918    SPITZER_SHA             cube …     PUBLIC           1  IRS-SH
      1746918    SPITZER_SHA             cube …     PUBLIC           1  IRS-SH
      1746918    SPITZER_SHA             cube …     PUBLIC           1  IRS-SH
      1746918    SPITZER_SHA             cube …     PUBLIC           1  IRS-SH
      1746918    SPITZER_SHA             cube …     PUBLIC           1  IRS-SH
      1746918    SPITZER_SHA             cube …     PUBLIC           1  IRS-SH
      1746918    SPITZER_SHA            image …     PUBLIC           1      --
      1746918    SPITZER_SHA            image …     PUBLIC           1      --
      1746918    SPITZER_SHA         spectrum …     PUBLIC           2  IRS-SH
          …            …              … …        …           …       …
      1746918    SPITZER_SHA         spectrum …     PUBLIC           2  IRS-SH
      1746918    SPITZER_SHA         spectrum …     PUBLIC           2  IRS-SH
```

```
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
 1746918    SPITZER_SHA         spectrum …    PUBLIC           2  IRS-SH
```

[21]:
```
obs1 = Observations.download_products('295913', productType="SCIENCE")
```

```
Downloading URL https://mast.stsci.edu/api/v0.1/Download/file?uri=mast:IUE/url/p
ub/iue/data/lwr/02000/lwr02252.mxlo.gz to
./mastDownload/IUE/lwr02252/lwr02252.mxlo.gz … [Done]
 [Done]
INFO: Found cached file ./mastDownload/IUE/lwr02252/lwr02252mxlo_vo.fits with
expected size 48960. [astroquery.query]
```

[22]:
```
obs1
```

[22]:
```
<Table length=2>
                   Local Path                    Status  Message  URL
                     str48                        str8    object object
------------------------------------------------ -------- -------- ------
    ./mastDownload/IUE/lwr02252/lwr02252.mxlo.gz COMPLETE     None   None
./mastDownload/IUE/lwr02252/lwr02252mxlo_vo.fits COMPLETE     None   None
```

1.0.3 Catalogs

[23]:
```
catalog_data = Catalogs.query_object("IC 418")
```

[24]:
```
catalog_data
```

[24]:
```
<Table masked=True length=2602>
 MatchID         Distance         …    W2_F658N_MAD     W2_F658N_N
  str9           float64          …      float64          int64
--------- ---------------------- … ------------------ ----------
 90374054 0.0024195100745320595 … 0.3621950149536133          2
 85650262  0.011976049939482375 …                nan          0
 59074158  0.012807511497058832 …                nan          0
 49514032  0.016868456405597757 …                0.0          1
  3849061   0.02201639285139814 …                nan          0
 79528152  0.025527337955486177 …                0.0          1
 34439260  0.028276258555586024 …                0.0          1
 60760708  0.032571346344362324 …                0.0          1
 14474603   0.03545828567463777 …                nan          0
      …                     … …                  …          …
 46442569     6.163894540712083 …                nan          0
 87561287    6.2027311621484245 …                nan          0
 40311846     6.203098083469302 …                nan          0
 98018064     6.211493316222754 …                nan          0
 24343268     6.243147156375478 …                nan          0
 33286019     6.279836860966224 …                nan          0
 13146491      6.36624878774303 …                nan          0
103027787     6.379484953628598 …                nan          0
 69478781     6.429983914053197 …                nan          0
 74166488      6.43736824292383 …                nan          0
```

1.1 Quasar images

From https://www.opencadc.org/notebook-tutorials/Notebooks/astroquery_example_quasars.html

See also: https://skyserver.sdss.org/dr14/en/tools/search/sql.aspx

[ ]:

```python
from urllib.parse import urlencode

from astropy.table import Table

# Define the query
query = """SELECT name, DR14_PLUG_RA AS RA, DR14_PLUG_DEC AS DEC,
    MJD, redshift,
    CLASS_BEST, CONF_BEST
FROM spiders_quasar
WHERE ( redshift > {redshift_lower}
    AND redshift < {redshift_upper}
    AND CLASS_BEST='QSO'
    AND DR14_ZWARNING=0 )""".format(redshift_lower=1.3, redshift_upper=1.5)

# Build the request
base_url = 'http://skyserver.sdss.org/dr14/SkyServerWS/SearchTools/SqlSearch'
parameters = {'cmd': query, 'format': 'fits'}
url_params = urlencode(parameters)
data_url = '{}?{}'.format(base_url, url_params)

# Grab the data from the url and convert to a pandas dataframe
qso_table = Table.read(data_url, format='fits')
qso_data = qso_table.to_pandas()

# Convert byte strings to regular strings
qso_data["CLASS_BEST"] = qso_data["CLASS_BEST"].str.decode("utf-8")
qso_data["name"] = qso_data["name"].str.decode("utf-8")

print('Number of results: {}'.format(len(qso_data)))
qso_data.head()
```

```
Number of results: 101
```

[ ]:

```
                    name         RA        DEC    MJD  redshift CLASS_BEST  \
0  2RXS J001129.4+005801   2.876672   0.964388  55478  1.490000        QSO
1  2RXS J003605.5+183808   9.026879  18.633067  56903  1.470117        QSO
2  2RXS J011517.4-012653  18.821264  -1.451258  56980  1.371068        QSO
3  2RXS J023154.7-045201  37.973090  -4.864214  56603  1.320000        QSO
4  2RXS J023507.3-040208  38.780584  -4.034806  57336  1.442799        QSO

   CONF_BEST
0          3
1          3
2          3
3          3
4          3
```

1.2 SDSS DR18, joining tables

[53]:

```python
# Define the query
query = """SELECT TOP 10000
    p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z,
    p.run, p.rerun, p.camcol, p.field,
    s.specobjid, s.class, s.z as redshift,
    s.plate, s.mjd, s.fiberid
FROM PhotoObj AS p
JOIN SpecObj AS s ON s.bestobjid = p.objid
WHERE
 p.u BETWEEN 0 AND 19.6
 AND g BETWEEN 0 AND 20"""

# Build the request
base_url = 'http://skyserver.sdss.org/dr18/SkyServerWS/SearchTools/SqlSearch'
parameters = {'cmd': query, 'format': 'fits'}
url_params = urlencode(parameters)
data_url = '{}?{}'.format(base_url, url_params)

# Grab the data from the url and convert to a pandas dataframe
dr18_table = Table.read(data_url, format='fits')
dr18_data = dr18_table.to_pandas()

dr18_data["class"] = dr18_data["class"].str.decode("utf-8")
dr18_data["specobjid"] = dr18_data["specobjid"].str.decode("utf-8")

print('Number of results: {}'.format(len(dr18_data)))
dr18_data.head()
```

```
Number of results: 10000
```

```
[53]:                  objid          ra        dec          u          g  \
      0  1237648721208672404  129.607617  -0.158606  19.062180  18.000385
      1  1237659326031201332  250.412780  36.190472  19.445974  18.480410
      2  1237658206122868827  187.712614  51.202387  18.920990  19.153826
      3  1237655369284648998  168.663323  60.905082  17.947710  17.373198
      4  1237659326033428565  254.251740  32.189918  18.696894  17.674751

                 r          i          z   run  rerun  camcol  field  \
      0  17.653460  17.666698  17.498667   756    301       3     92
      1  18.176233  18.054541  18.061897  3225    301       4    266
      2  19.581593  19.934433  20.104891  2964    301       6    333
      3  17.597744  17.377859  17.386299  2304    301       2    145
      4  17.329300  17.189348  17.096918  3225    301       4    300

                    specobjid   class  redshift  plate    mjd  fiberid
      0   8390304900123482112    STAR  0.000010   7452  56745      359
      1   2460136985440918528    STAR -0.000766   2185  53532      166
      2   7514391050963933184    STAR  0.000177   6674  56416      491
      3    872652455777691648  GALAXY  0.053536    775  52295      291
      4  12282679717174466560    STAR -0.000363  10909  58280      864
```

- DR8 handles searches by spectral lines differently than previous releases. In addition, spectral lines for galaxies and stars are
- identified through different processes.
- Spectral lines for galaxies are calculated using the MPA-JHU spectroscopic reanalysis (Tremonti et al. 2004; Brinchmann et al. 2004) and are stored in the galSpecLine table. For more on how spectral lines of galaxies are found, see the Galspec page of the sdss3.org website.
- Spectral lines for stars are calculated using the SEGUE Stellar Parameter Pipeline (SSPP; Lee et al. 2008) and are stored in the sppLines
- table.
- For more on how spectral lines of stars are found, see the SSPP page of the sdss3.org website.
- Finding galaxies by their emission lines:

This query selects galaxy spectra with high internal reddening, as measured by the standard Balmer decrement technique. It makes use of the galSpec tables for the measurements of galaxy lines. In this case we use galSpecLine, which has emission line measurements.

```
[51]: Hab_min = 8.
      query = f""" SELECT
      s.plate, s.fiberid, s.mjd, s.z, s.zwarning,
      g.h_beta_flux, g.h_beta_flux_err,
      g.h_alpha_flux, g.h_alpha_flux_err
      FROM GalSpecLine AS g
      JOIN SpecObj AS s
```

```
ON s.specobjid = g.specobjid
WHERE
h_alpha_flux > h_alpha_flux_err*5
AND h_beta_flux > h_beta_flux_err*5
AND h_beta_flux_err > 0
AND h_alpha_flux > {Hab_min} * h_beta_flux
AND s.class = 'GALAXY'
AND s.zwarning = 0"""

# Build the request
base_url = 'http://skyserver.sdss.org/dr18/SkyServerWS/SearchTools/SqlSearch'
parameters = {'cmd': query, 'format': 'fits'}
url_params = urlencode(parameters)
data_url = '{}?{}'.format(base_url, url_params)

# Grab the data from the url and convert to a pandas dataframe
gals_table = Table.read(data_url, format='fits')
gals_data = gals_table.to_pandas()

print('Number of results: {}'.format(len(gals_data)))
gals_data.head()
```

```
Number of results: 1297
```

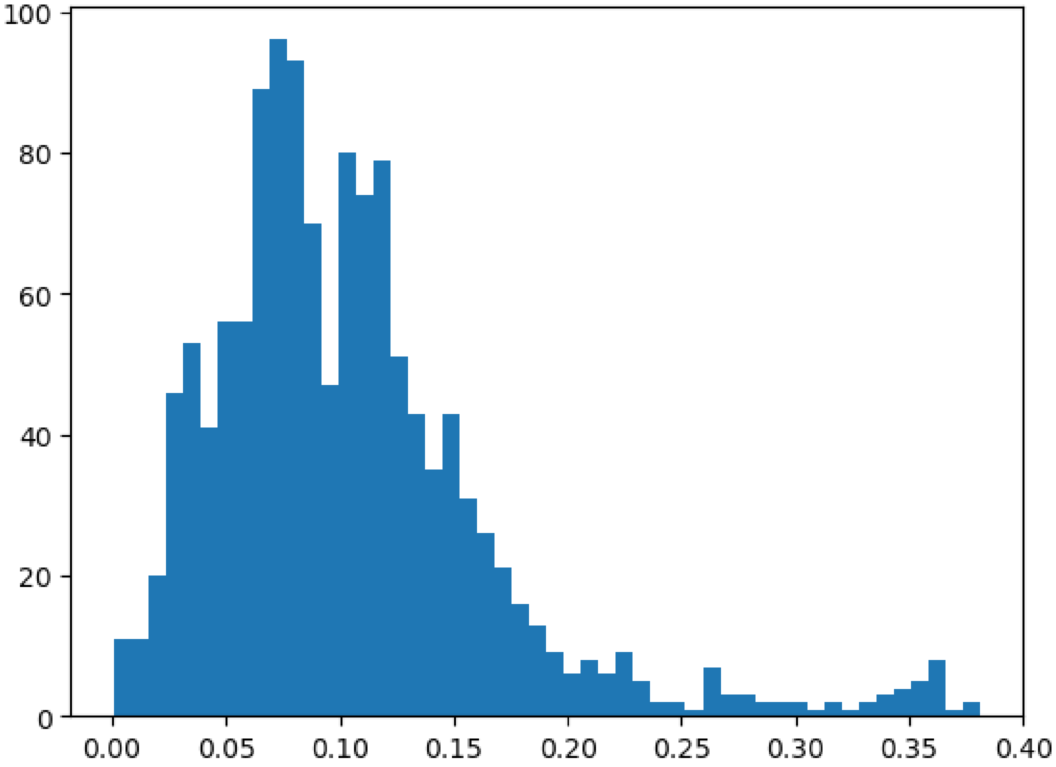
[51]:

```
   plate  fiberid    mjd         z  zwarning  h_beta_flux  h_beta_flux_err  \
0   1023      464  52818  0.213886         0    17.707502         2.047773
1    990       91  52465  0.138776         0    16.826162         3.283018
2    387      607  51791  0.139045         0    16.839197         2.852956
3   2086      149  53401  0.231597         0    23.869757         3.411085
4   1716      265  53827  0.116116         0    13.177597         2.533214

   h_alpha_flux  h_alpha_flux_err
0    155.332977          4.939080
1    155.304245          4.398665
2    139.462814          3.978872
3    215.193619          5.971058
4    111.353470          3.981305
```

[52]:

```
f, ax = plt.subplots()
ax.hist(gals_data['z'], bins=50);
```


100
80
60
40
20
0
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40

[ ]: