tutorial1

March 26, 2023

1 Tutorial 1: SDSS Query

Instructor: Isidora Jankov

Student: Ridha Fathima Mohideen Malik

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
from astroquery.sdss import SDSS
from astroquery.sdss import SDSS
```

1.1 Querying on Python

1. Select galaxies and quasars with redshifts between 0.05 and 0.3 and signalto-noise ratios greater than 35 near the H line. Ensure that lines [O III] 5007, H 4863, and H 4341 are present in emission and that the FWHM of H is greater than 1000 km/s. For each selected spectrum, find the flux ratios of [OIII]/H , H /H , and [O III]/H , as well as the equivalent width and flux of H , redshift, and extinction correction: E(B-V) of type SFD (tip: the last one find in galSpecInfo table).

I used the following query using https://skyserver.sdss.org/dr18/SearchTools/sql:

```
s.plate, s.mjd, s.fiberid,
s.z, g.subclass, g.e_bv_sfd,
l.h_beta_flux, l.h_beta_eqw,
(l.oiii_5007_flux/l.h_beta_flux) as oiii_h_beta_flux_ratio,
(l.oiii_5007_flux/l.h_gamma_flux) as oiii_h_gamma_flux_ratio,
(l.h_beta_flux/l.h_gamma_flux) as h_beta_h_gamma_flux_ratio

FROM SpecObjAll AS s
JOIN GalSpecInfo AS g ON s.specobjid = g.specobjid
JOIN GalSpecLine AS l ON s.specobjid = l.specobjid

WHERE
(s.class = "QSO" OR s.class = "GALAXY")
AND s.z BETWEEN 0.05 AND 0.6
AND s.snmedian_g > 35
AND l.h_gamma_eqw < 0
AND l.h_beta_eqw < 0
AND l.oiii_5007_eqw < 0
```

```
AND l.h_gamma_flux <> 0
AND l.h_beta_flux <> 0
AND l.oiii_5007_flux <> 0
AND l.sigma_balmer * 2.35 > 1000
```

Alternatively, this could be done with python as follows.

Resulting table in *420primary.csv*

[ ]:
```
with open('query.txt', 'r') as file:
    query = file.read().replace('\n', '') # input is a single string

res = SDSS.query_sql(query)
res # output is an astropy table
```

[ ]:
```
<Table length=420>
plate  mjd  fiberid … oiii_h_gamma_flux_ratio h_beta_h_gamma_flux_ratio
int32 int32  int32  …         float64                   float64
----- ----- ------- … ----------------------- -------------------------
  390 51900     587 …                1.645157                  1.677394
  355 51788     414 …                1.879482                  2.642485
 1233 52734     312 …                2.169284                  1.449437
 2227 53820     252 …                3.758329                  1.402163
  519 52283     280 …                2.025708                  1.964605
 2156 54525     122 …                1.029048                  2.071693
 2647 54495     378 …                1.048939                 0.8033115
  494 51915     124 …                3.919306                  2.127565
  607 52368     581 …                1.061532                  2.359785
 1673 53462     108 …                1.353253                   2.23479
  …   …      … …                      …                         …
 2646 54479     204 …               0.2995124                  1.913876
  464 51908     576 …               0.4592269                  1.133334
 1579 53473      88 …                1.564971                  1.260422
 1819 54540     140 …                1.648178                  2.114187
 2520 54584     442 …                 6.07903                  1.718534
 2142 54208     637 …                1.380025                  2.219383
 2266 53679     335 …                1.565774                  2.036093
 2155 53820     341 …               0.4473922                  1.768658
 2202 53566     592 …               0.5782274                  2.077073
  437 51876     234 …                2.333672                   1.87826
```

2. How many objects have you found? Which one from the conditions in WHERE is narrowing the results most severely? (TIP: one needs to play with this for a while…)

condition	count
all	420
z removed	1621
sn removed	35502
fwhm 500	467

condition	count
fwhm removed	623

The signal-to-noise ratio condition has the most impact on the result.

3. Find out if there is some of the Subclass AGN objects, with the same conditions under 1. Adopt your code to get result.

Yes, Broadline AGNs

1.2 Crossmatching

4. Using the problem solution under 1 and the list of objects (287-plate-mjdfiber.txt) submit the SQL query via CrossID. (TIP: you will need to alter the SQL code prepared under 1 to fit requirements of CrossID. Follow the comments you get and be patient)

```
CREATE TABLE #upload ( up_id int, up_plate int, up_mjd int, up_fiber int )
INSERT INTO #upload values ( 1, 1949, 53433, 472),( 2, 1273, 52993, 348),( 3, 2030, 53499, 201)
SELECT
s.plate, s.mjd, s.fiberid,
s.z, g.subclass, g.e_bv_sfd,
l.h_beta_flux, l.h_beta_eqw,
(l.oiii_5007_flux/l.h_beta_flux) as oiii_h_beta_flux_ratio,
(l.oiii_5007_flux/l.h_gamma_flux) as oiii_h_gamma_flux_ratio,
(l.h_beta_flux/l.h_gamma_flux) as h_beta_h_gamma_flux_ratio

FROM #upload u
JOIN
SpecObjAll AS s
ON (s.plate=u.up_plate AND s.mjd=u.up_mjd AND s.fiberID=u.up_fiber)
JOIN GalSpecInfo AS g ON s.specobjid = g.specobjid
JOIN GalSpecLine AS l ON s.specobjid = l.specobjid

WHERE
(s.class = "QSO" OR s.class = "GALAXY")
AND s.z BETWEEN 0.05 AND 0.6
AND s.snmedian_g > 35
AND l.h_gamma_eqw < 0
AND l.h_beta_eqw < 0
AND l.oiii_5007_eqw < 0
AND l.h_gamma_flux > 0
AND l.h_beta_flux > 0
AND l.oiii_5007_flux > 0
AND (l.sigma_balmer * 2.35) > 1000
```

This produced 47 rows. Results in *47crossmatch.csv*

I rechecked with pandas:

[ ]:
```
given = pd.read_csv("287-plate-MJD-fiberID.txt", sep=r'\s+|\t',␣
↪engine='python', header=None)
given.head()
```

[ ]:

	redshift	h_beta_eqw	h_beta_flux	Column1	Column2	Column3	\
0	0.599349	-11.40223	3992.1940	0.252373	2.374881	0.599357	
1	0.434385	-15.68500	1279.6670	0.646228	2.294816	1.482974	
2	0.468860	-10.00354	552.9328	0.637080	2.148320	1.368653	
3	0.468984	-20.90205	1784.6600	0.186969	2.345961	0.438622	
4	0.435857	-11.22126	178.4678	3.537334	1.718534	6.079030	

	spectofiber	plate	mjd	fiberID
0	0.731025	637	52174	259
1	0.748482	1754	53385	324
2	0.938292	555	52266	74
3	0.784193	976	52413	574
4	0.281381	2520	54584	442

[ ]:
```
got = pd.read_csv("420primary.csv", sep=',', header=0, comment='#')
got.head()
```

[ ]:

	plate	mjd	fiberid	z	subclass	e_bv_sfd	h_beta_flux	\
0	300	51666	135	0.599775	BROADLINE	0.039406	2747.5760	
1	412	52258	129	0.080176	BROADLINE	0.065049	2925.6050	
2	554	52000	553	0.473671	BROADLINE	0.025234	961.0248	
3	555	52266	74	0.468860	BROADLINE	0.021617	552.9328	
4	1012	52649	74	0.102776	BROADLINE	0.012979	3461.1010	

	h_beta_eqw	oiii_h_beta_flux_ratio	oiii_h_gamma_flux_ratio	\
0	-21.376590	0.398863	0.874091	
1	-12.630870	0.559593	1.125262	
2	-8.415974	1.069571	1.917614	
3	-10.003540	0.637080	1.368653	
4	-22.343320	0.722970	1.699948	

	h_beta_h_gamma_flux_ratio
0	2.191457
1	2.010858
2	1.792881
3	2.148320
4	2.351340

[ ]:
```
for id, plate_given, mjd_given, fiber_given in zip(given.index, given['plate'],␣
↪given['mjd'], given['fiberID']):
        for plate_got, mjd_got, fiber_got in zip(got['plate'], got['mjd'],␣
↪got['fiberid']):
```

```
                if plate_given == plate_got and mjd_given == mjd_got and␣
↪fiber_given == fiber_got:
                        given.loc[id, 'found'] = "yes"
```

[ ]:
```
len(given[given['found']=="yes"])
```

[ ]: 47

[ ]:
```
result = pd.read_csv("47crossmatch.csv", sep=',', header=0, comment='#')
```

1.3 Understanding the spectrum file

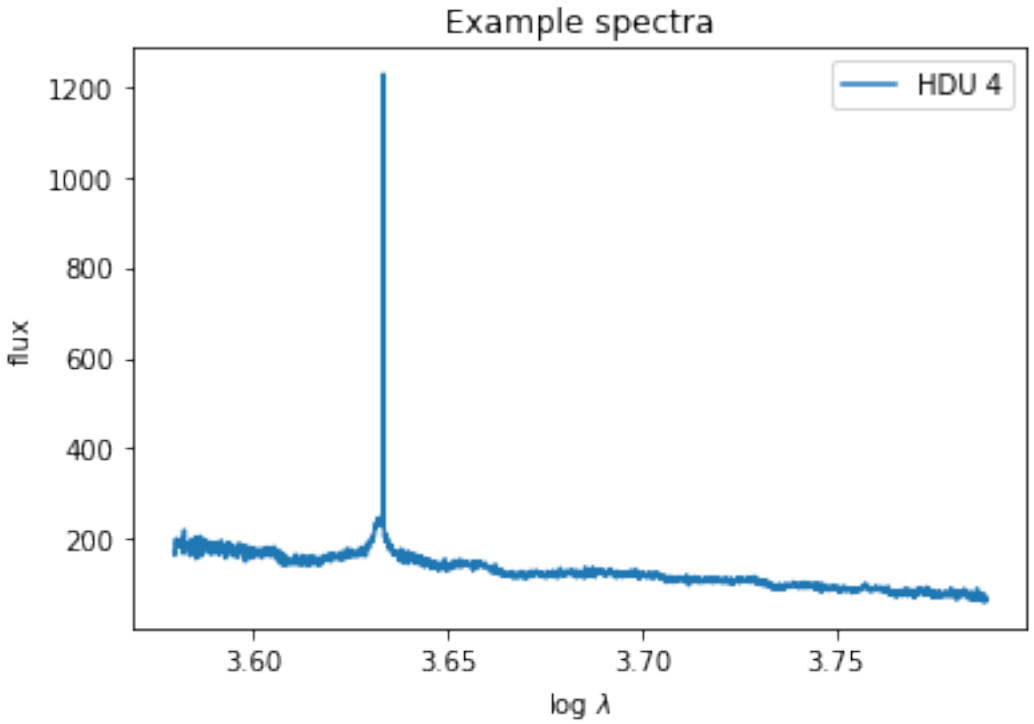
5. Check the spectra of found objects, download some of them using wget.
6. BONUS: read downloaded fits files and plot the spectra using Python.

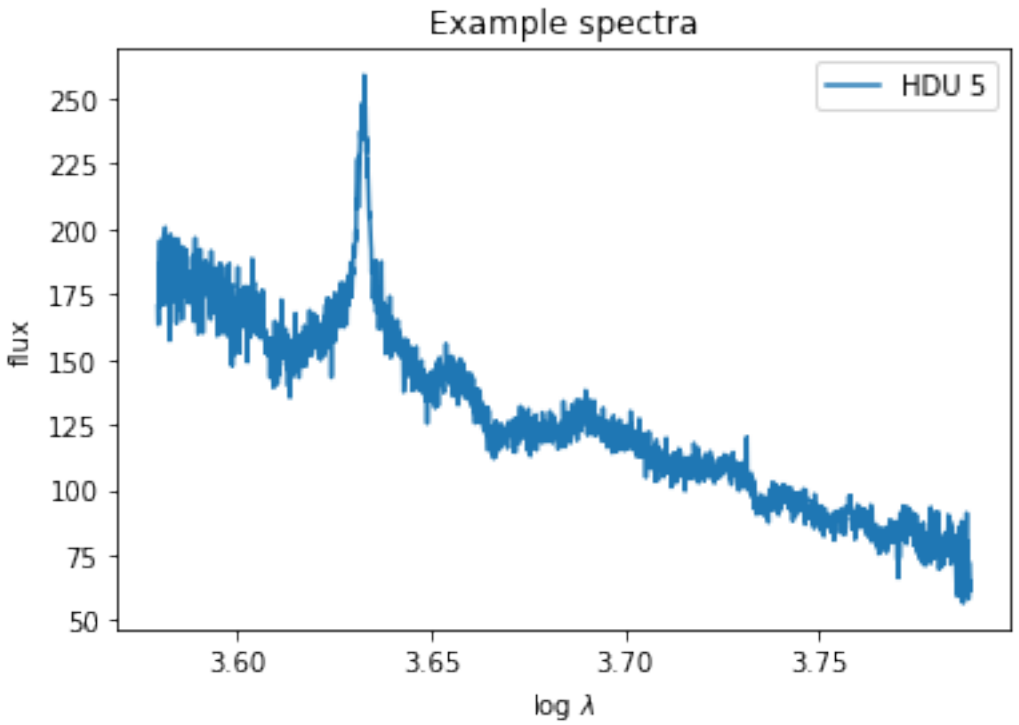
Since the SAS server was down, I explored other options to download spectra. Here I am employing astroquery to plot and understand the results.

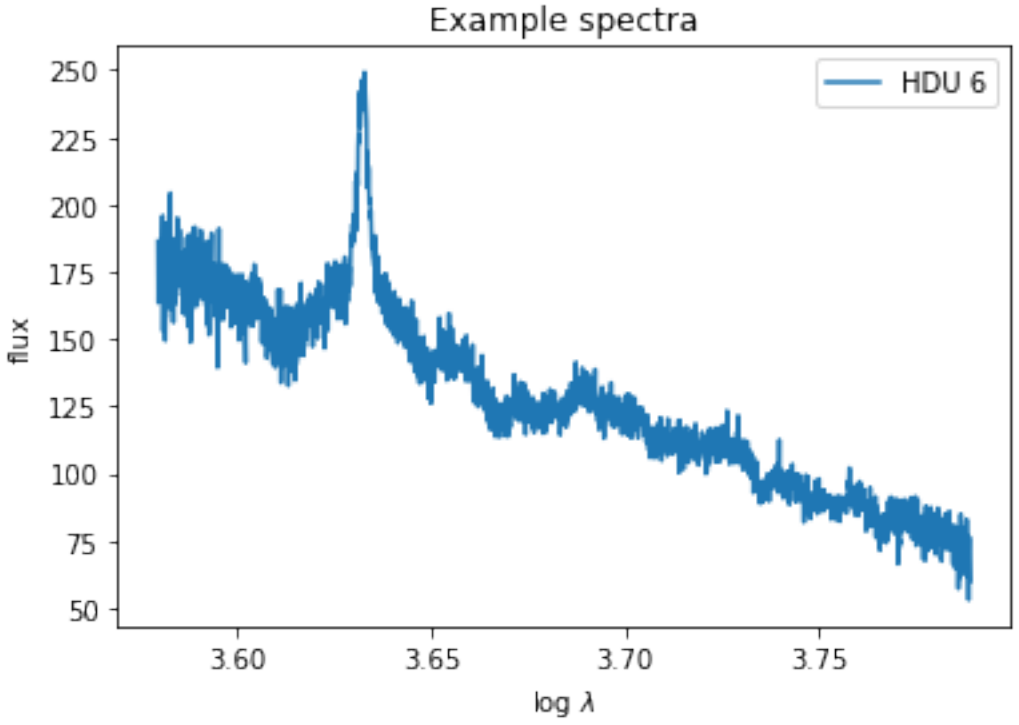
[ ]:
```
sp = SDSS.get_spectra(plate=result.loc[1,"plate"], mjd=result.loc[1,"mjd"],␣
↪fiberID=result.loc[1,"fiberid"]) # list of HDUs
sp[0].info()
```

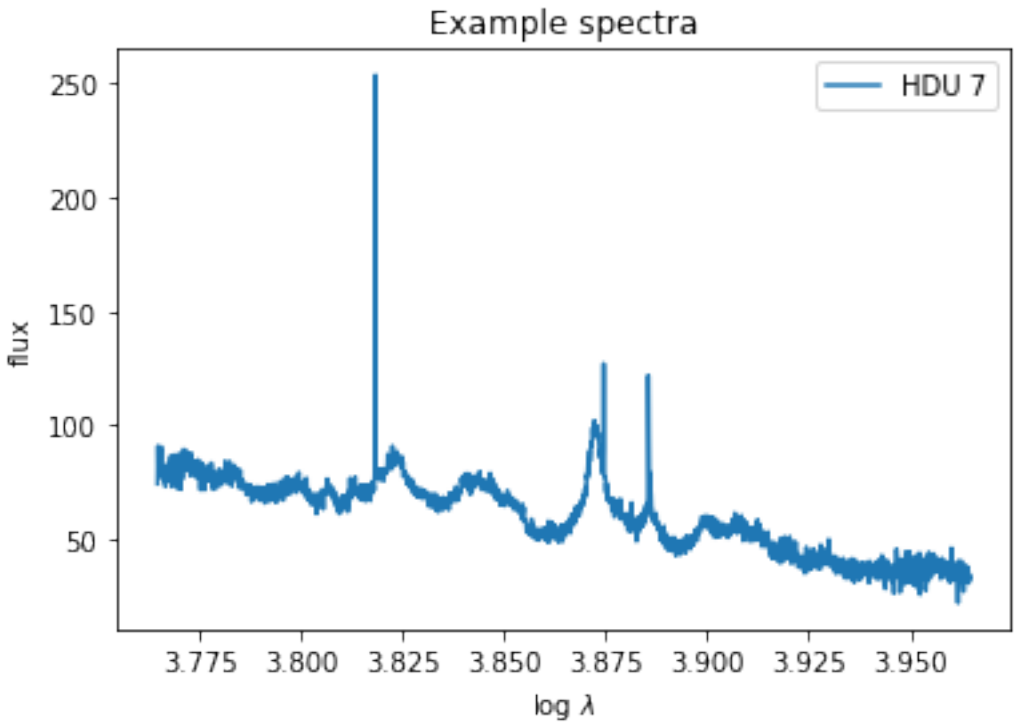
```
Filename: (No file associated with this HDUList)
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU     138   ()
  1  COADD         1 BinTableHDU     26   3848R x 8C   [E, E, E, J, J, E, E, E]
  2  SPECOBJ       1 BinTableHDU    262   1R x 126C   [6A, 4A, 16A, 23A, 16A,
8A, E, E, E, J, E, E, J, B, B, B, B, B, B, J, 22A, 19A, 19A, 22A, 19A, I, 3A,
3A, 1A, J, D, D, D, E, E, 19A, 8A, J, J, J, J, K, K, J, J, J, J, J, J, K, K, K,
K, I, J, J, J, J, 5J, D, D, 6A, 21A, E, E, E, J, E, 24A, 10J, J, 10E, E, E, E,
E, E, E, J, E, E, E, J, E, 5E, E, 10E, 10E, 10E, 5E, 5E, 5E, 5E, 5E, J, J, E, E,
E, E, E, E, 25A, 21A, 10A, E, E, E, E, E, E, E, E, J, E, E, J, 1A, 1A, E, E, J,
J, 1A, 5E, 5E]
  3  SPZLINE       1 BinTableHDU     48   29R x 19C   [J, J, J, 13A, D, E, E, E,
E, E, E, E, E, E, E, J, J, E, E]
  4  B2-00006798-00006802-00006803    1 BinTableHDU    145   2047R x 7C   [E, E,
E, J, E, E, E]
  5  B2-00006799-00006802-00006803    1 BinTableHDU    145   2047R x 7C   [E, E,
E, J, E, E, E]
  6  B2-00006800-00006802-00006803    1 BinTableHDU    145   2047R x 7C   [E, E,
E, J, E, E, E]
  7  R2-00006798-00006802-00006803    1 BinTableHDU    145   2044R x 7C   [E, E,
E, J, E, E, E]
  8  R2-00006799-00006802-00006803    1 BinTableHDU    145   2044R x 7C   [E, E,
E, J, E, E, E]
  9  R2-00006800-00006802-00006803    1 BinTableHDU    145   2044R x 7C   [E, E,
E, J, E, E, E]
```

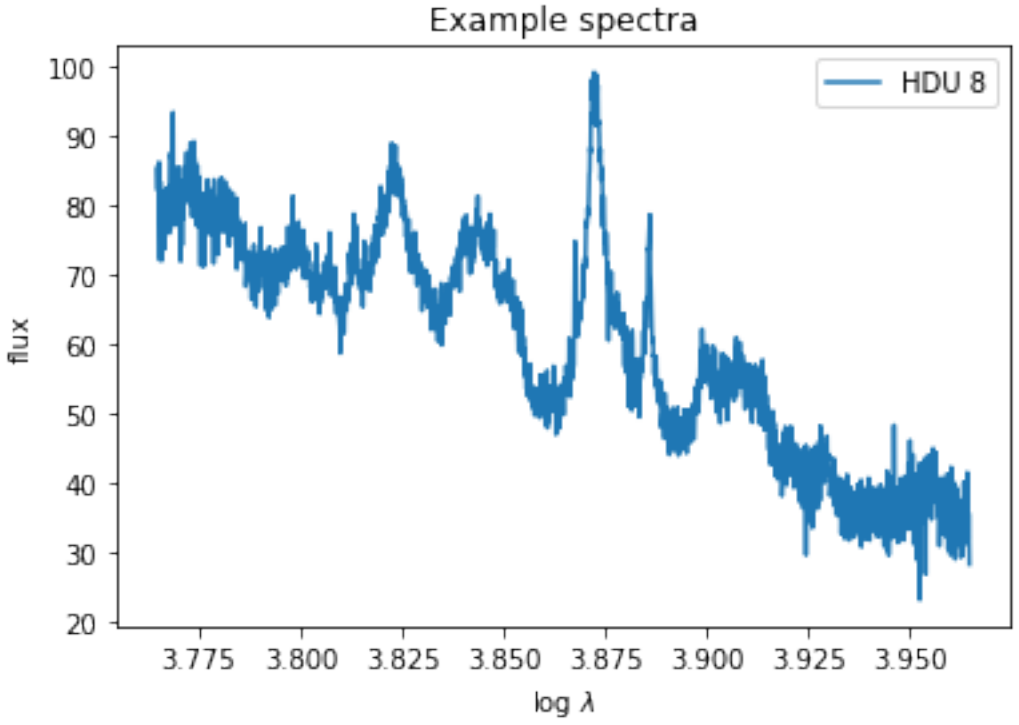
[ ]:
```
for i in range(4,10):
    spectrum = sp[0][i]
    plt.plot(spectrum.data['loglam'], spectrum.data['flux'], label='HDU␣
 ↪'+str(i))
    plt.xlabel(r'log $\lambda$')
    plt.ylabel('flux')
    plt.title(f"Example spectra")
    plt.legend()
    plt.show()
```

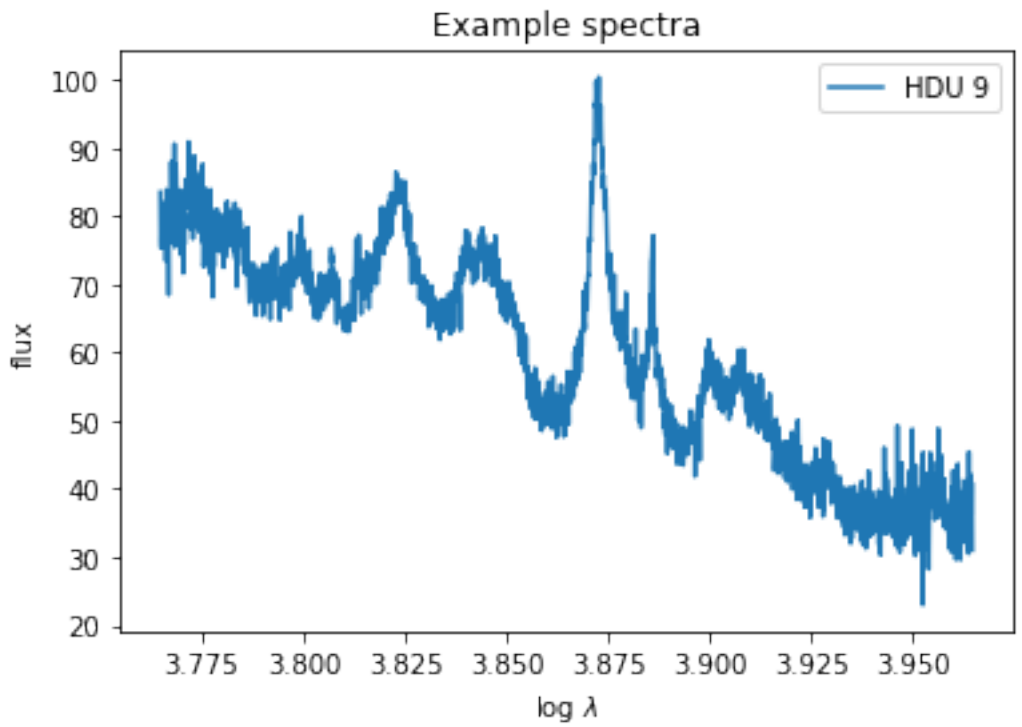



Example spectra
HDU 5
flux
250
225
200
175
150
125
100
75
50
3.60
3.65
3.70
3.75
log λ


Example spectra
HDU 6
flux
250
225
200
175
150
125
100
75
50
3.60
3.65
3.70
3.75
log λ


Example spectra
HDU 7
250
200
150
100
50
flux
3.775
3.800
3.825
3.850
3.875
3.900
3.925
3.950
log λ


Example spectra
HDU 8
100
90
80
70
60
50
40
30
20
flux
3.775
3.800
3.825
3.850
3.875
3.900
3.925
3.950
log λ



Observations: - HDUs 4-6 are the bluer wavelengths and 7-9 correspond to the redder wavelengths - HDUs 6 and 9 are the reduced results

1.3.1 Final 47 spectra

[ ]:
```
for i in range(len(result)):
    print("# ", i+1)
    plate=result.loc[i,"plate"]
    mjd=result.loc[i,"mjd"]
    fiberID=result.loc[i,"fiberid"]
    sp = SDSS.get_spectra(plate=plate, mjd=mjd, fiberID=fiberID)
    spectrum_b = sp[0][6]
    spectrum_r = sp[0][9]
    plt.plot(spectrum_b.data['loglam'], spectrum_b.data['flux'], label='B',␣
 ↪c='b')
    plt.plot(spectrum_r.data['loglam'], spectrum_r.data['flux'], label='R',␣
 ↪c='r')
    plt.xlabel(r'log $\lambda$')
    plt.ylabel('flux')
    plt.title(f"Plate: {plate} MJD: {mjd} FiberID: {fiberID}")
    plt.legend()
    plt.show()
```

1

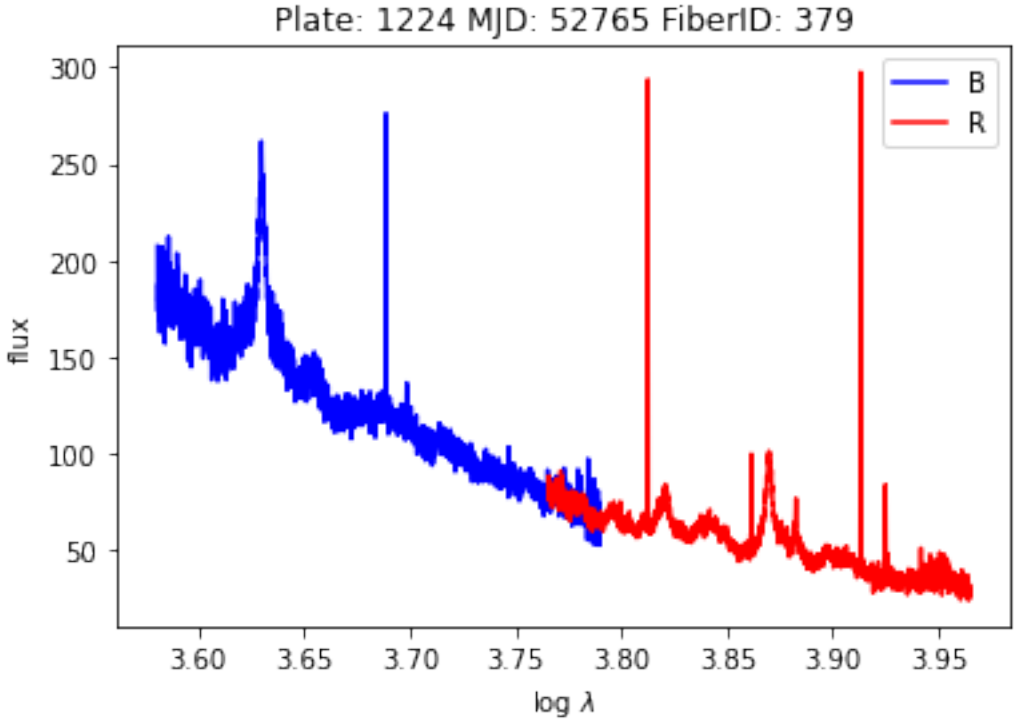


Plate: 1224 MJD: 52765 FiberID: 379
B
R
flux
300
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

2

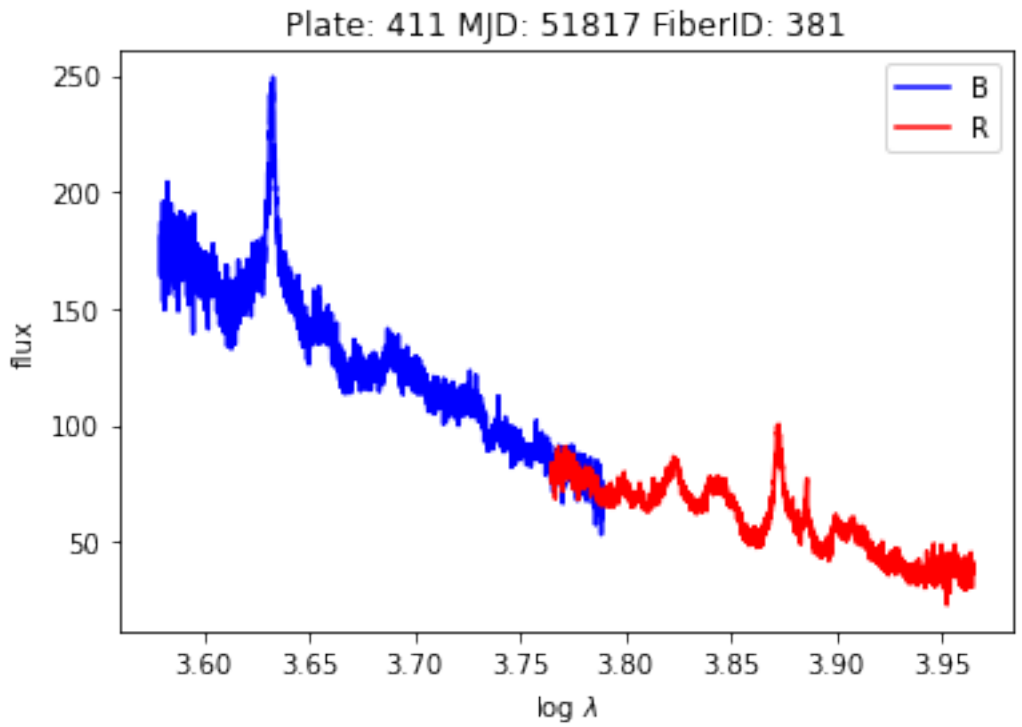


Plate: 411 MJD: 51817 FiberID: 381
B
R
flux
log λ
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

3

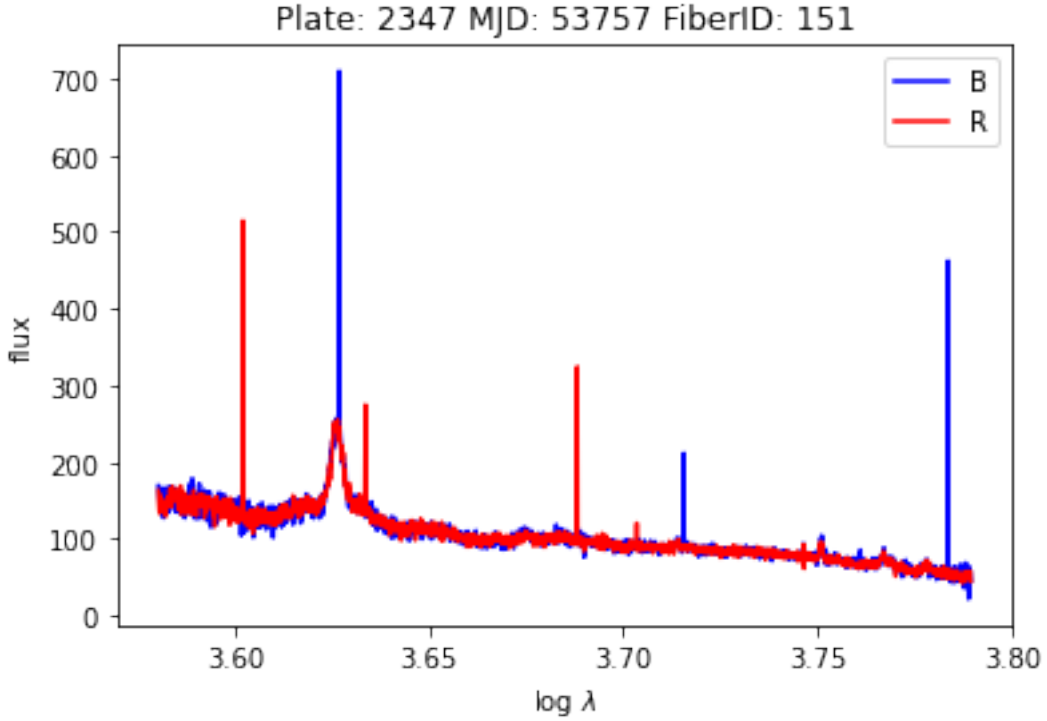


Plate: 2347 MJD: 53757 FiberID: 151
B
R
flux
log λ
700
600
500
400
300
200
100
0
3.60
3.65
3.70
3.75
3.80

```
#  4
```

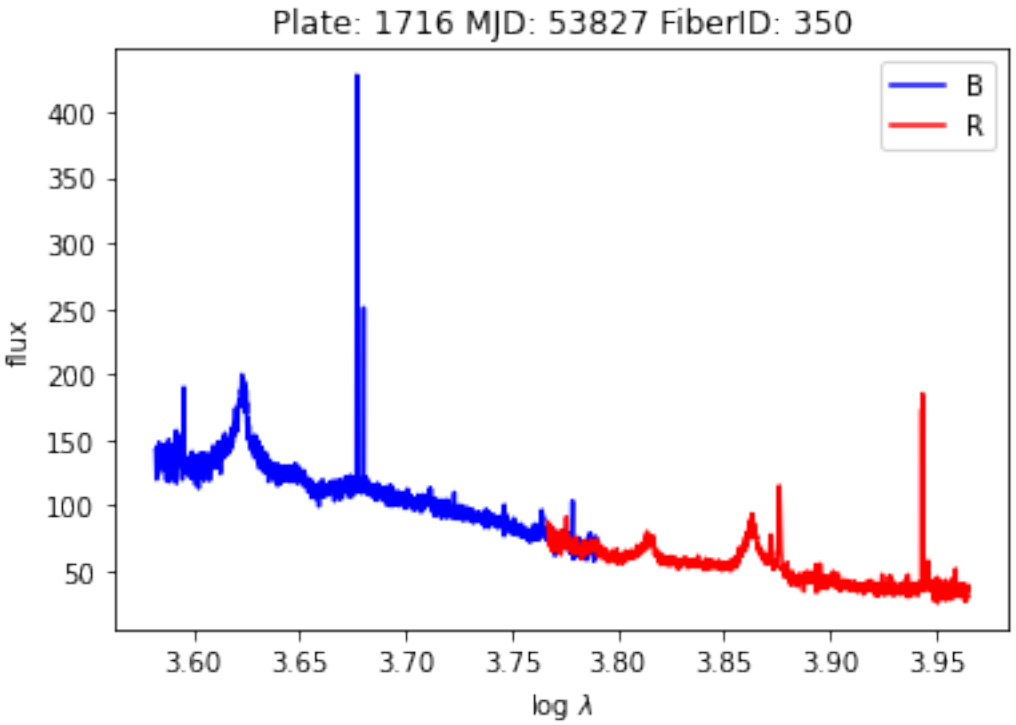


Plate: 1716 MJD: 53827 FiberID: 350
B
R
flux
400
350
300
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

```
#  5
```

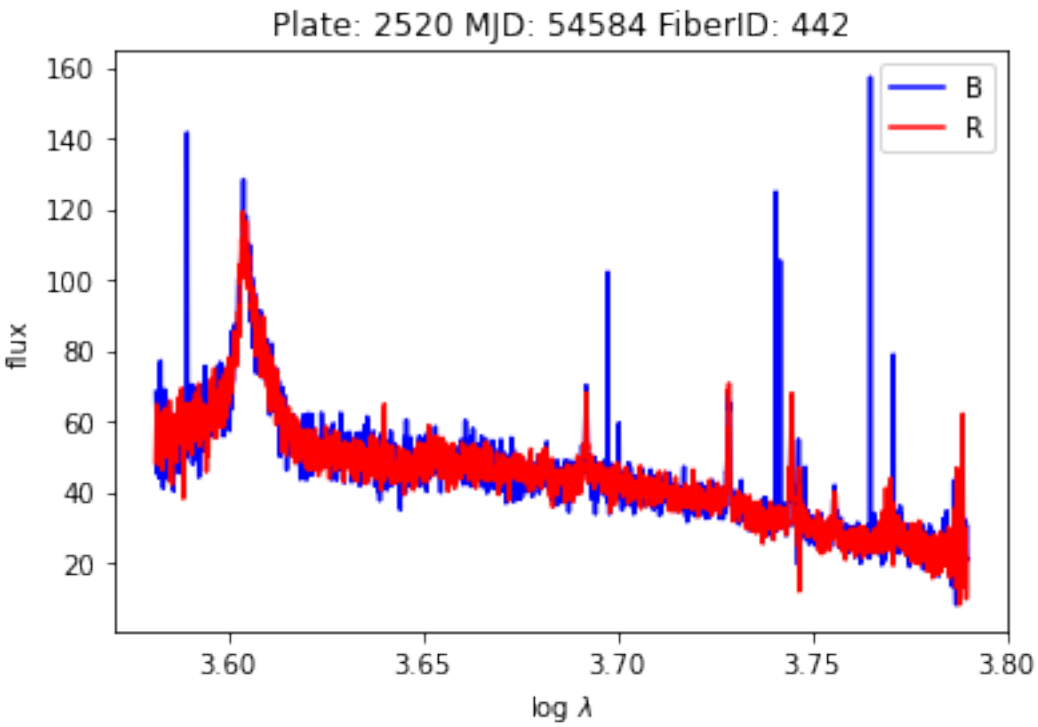


Plate: 2520 MJD: 54584 FiberID: 442
B
R
flux
log λ
160
140
120
100
80
60
40
20
3.60
3.65
3.70
3.75
3.80

6

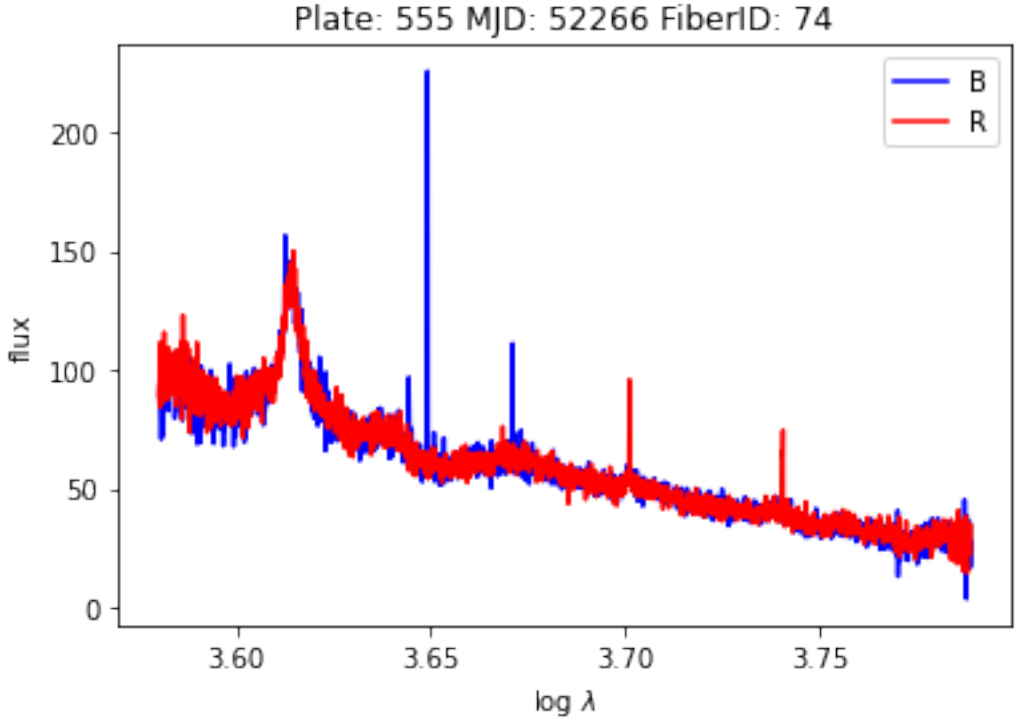


Plate: 555 MJD: 52266 FiberID: 74
B
R
flux
log λ
200
150
100
50
0
3.60
3.65
3.70
3.75

```
#  7
```

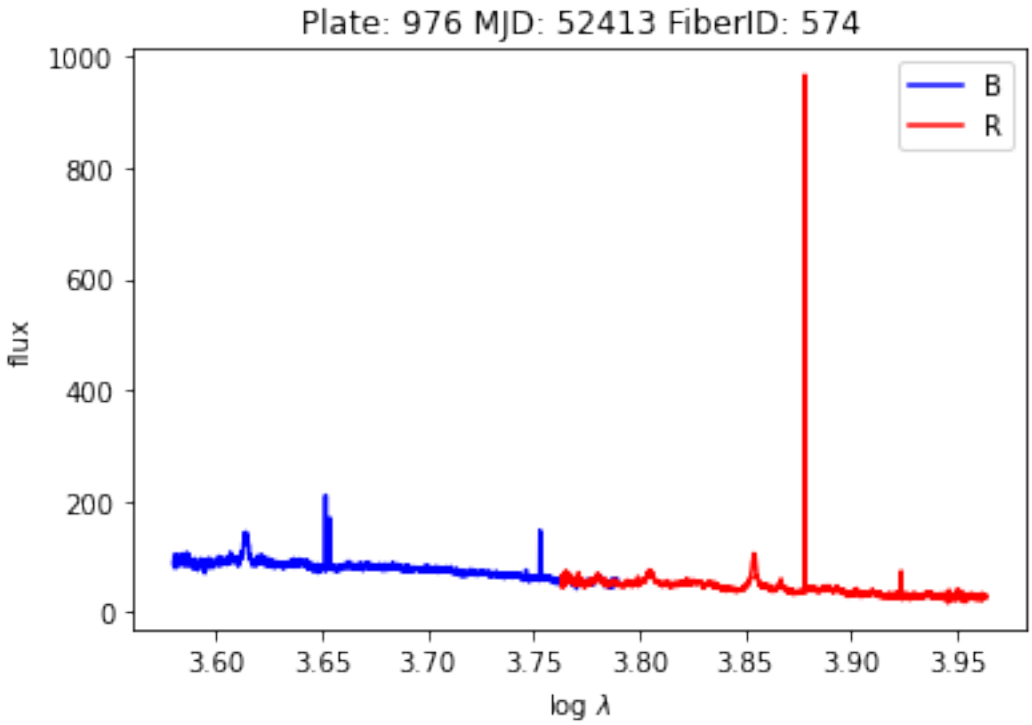


Plate: 976 MJD: 52413 FiberID: 574
B
R
flux
log λ
1000
800
600
400
200
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

```
#  8
```

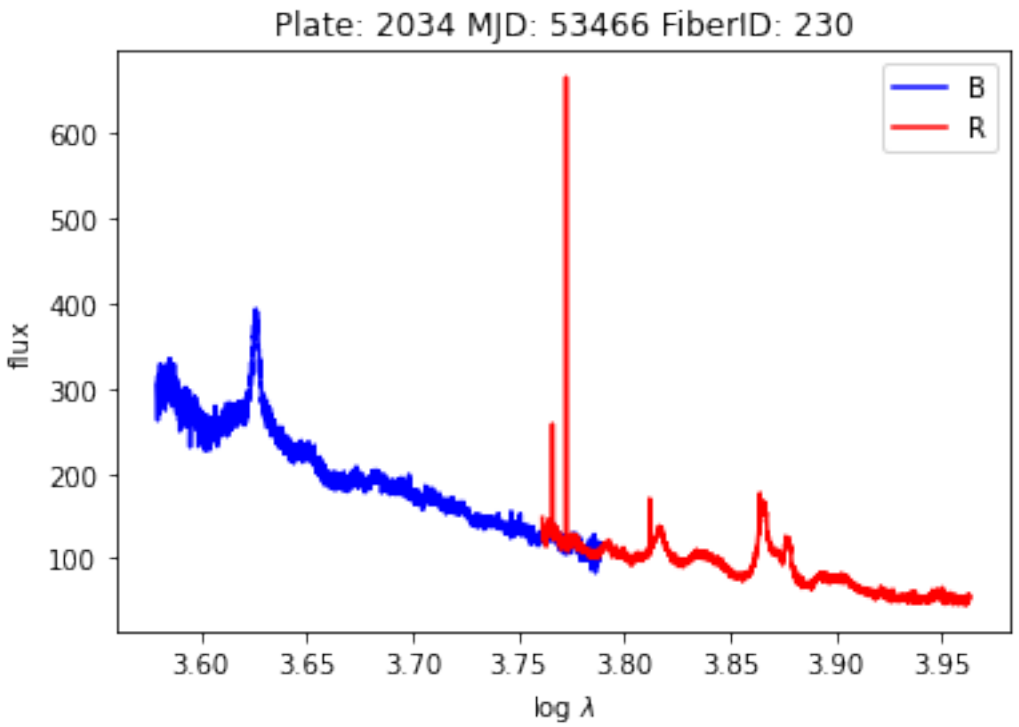


Plate: 2034 MJD: 53466 FiberID: 230
B
R
flux
600
500
400
300
200
100
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

9

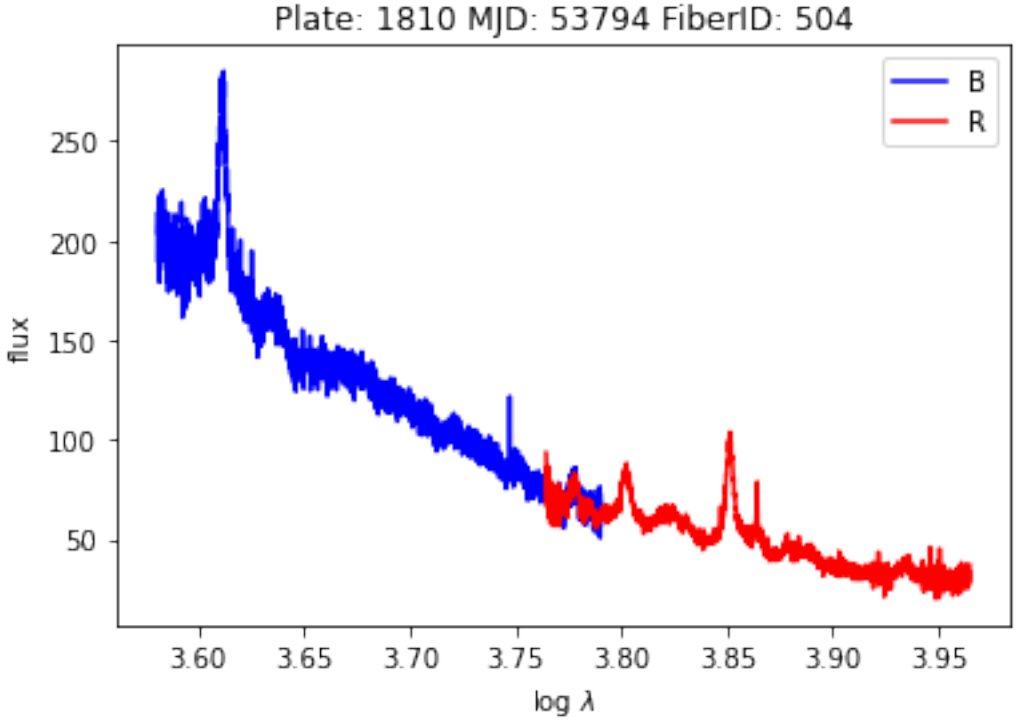


Plate: 1810 MJD: 53794 FiberID: 504
B
R
flux
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

```
#  10
```

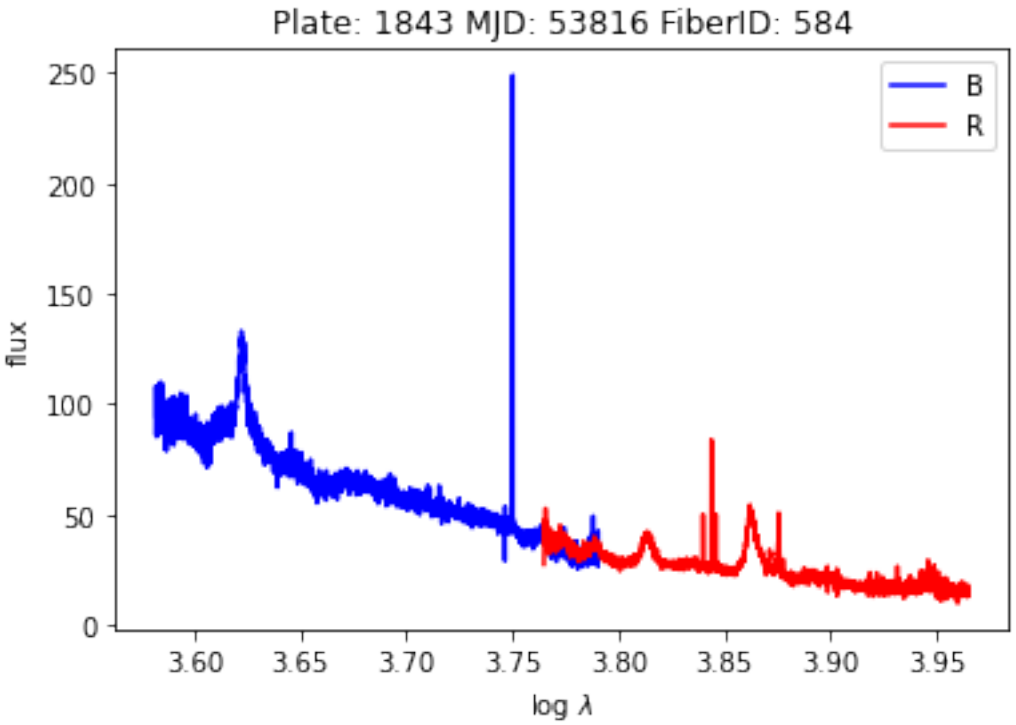


Plate: 1843 MJD: 53816 FiberID: 584
B
R
flux
log λ
250
200
150
100
50
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

```
#  11
```

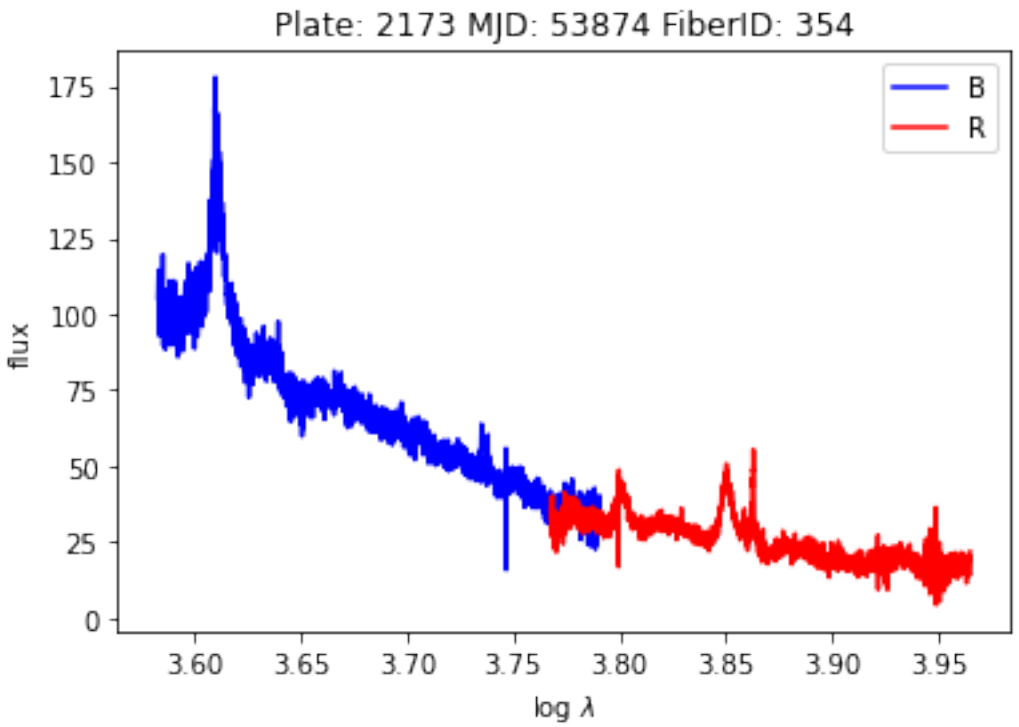


Plate: 2173 MJD: 53874 FiberID: 354
B
R
flux
log λ
0
25
50
75
100
125
150
175
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

```
#  12
```

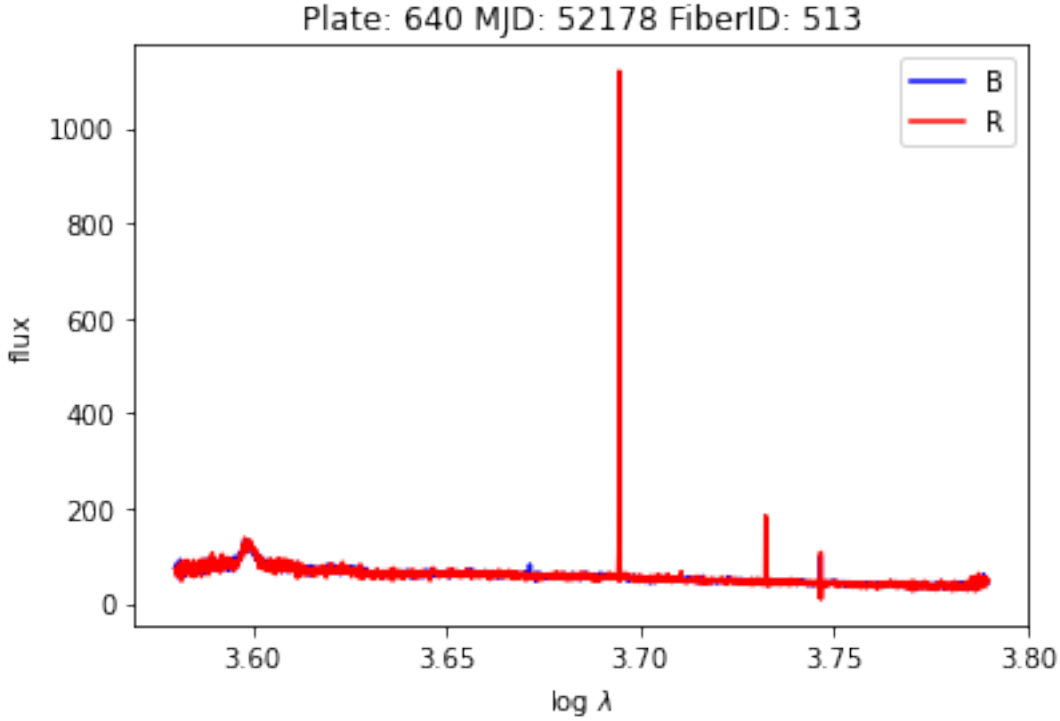


Plate: 640 MJD: 52178 FiberID: 513
B
R
flux
log λ
0
200
400
600
800
1000
3.60
3.65
3.70
3.75
3.80

```
#  13
```

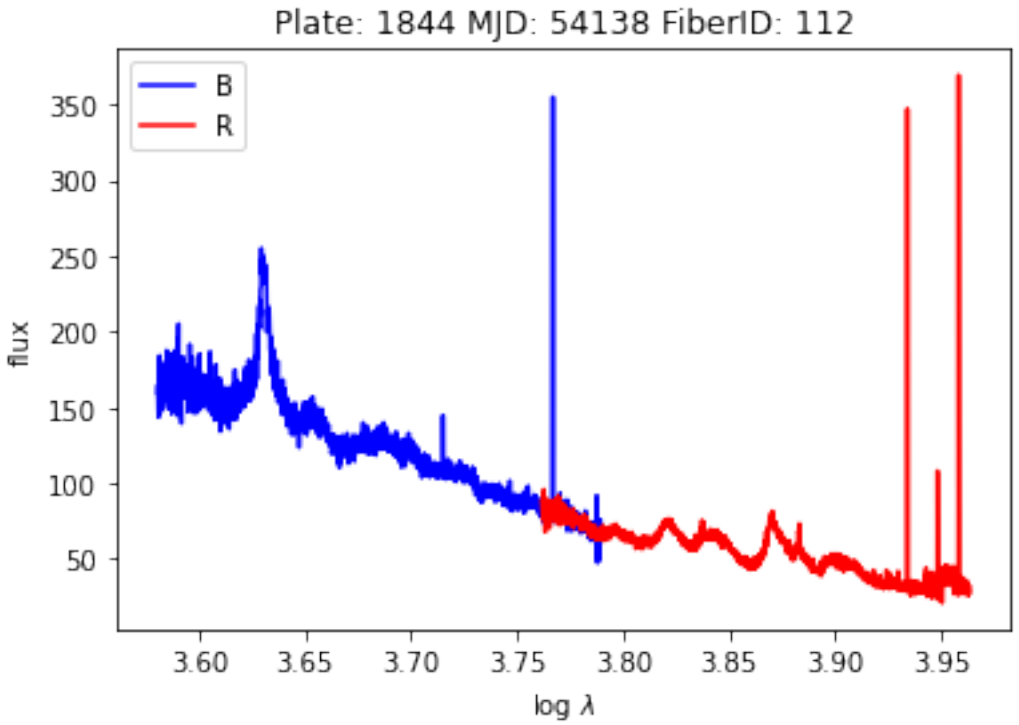


Plate: 1844 MJD: 54138 FiberID: 112
B
R
flux
350
300
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

```
#  14
```

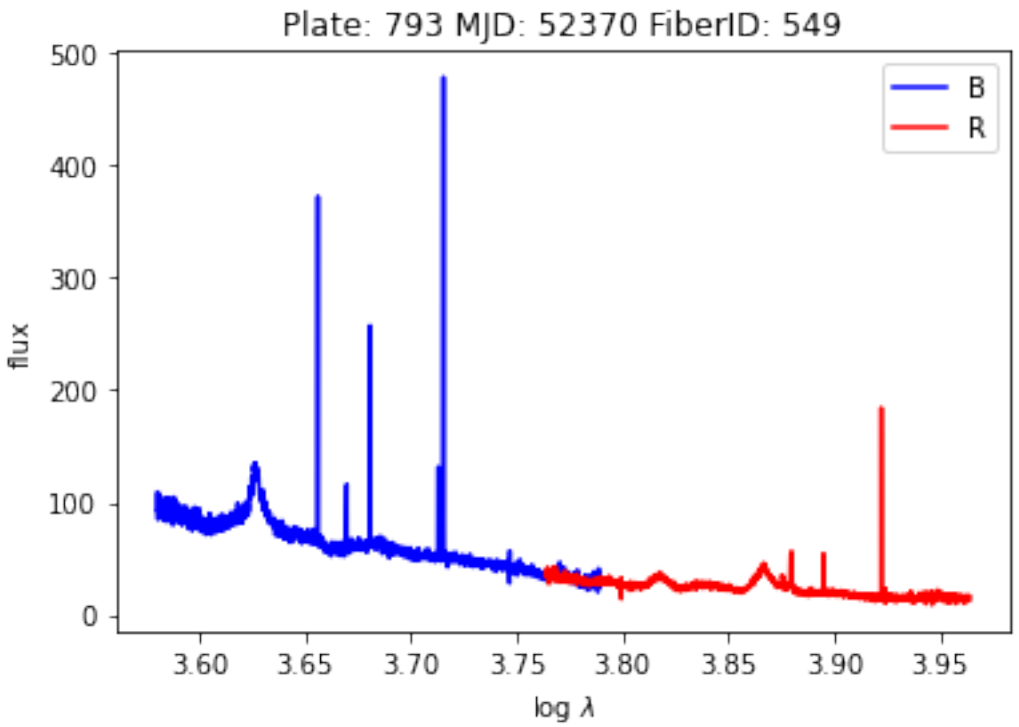


Plate: 793 MJD: 52370 FiberID: 549
B
R
flux
log λ
500
400
300
200
100
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

```
#  15
```

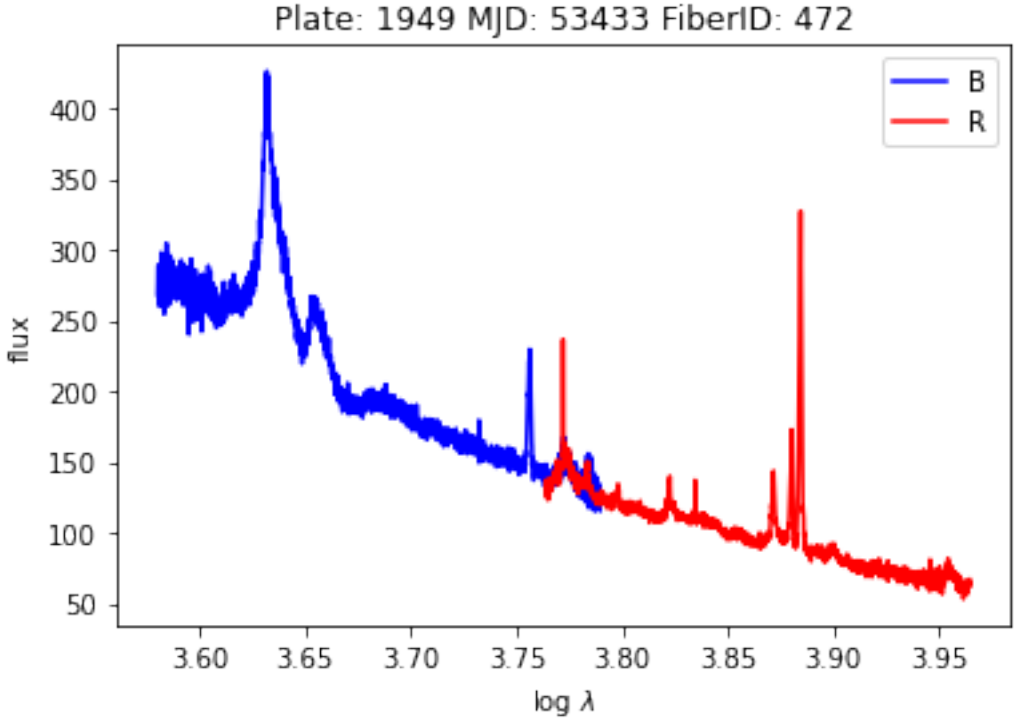


Plate: 1949 MJD: 53433 FiberID: 472
B
R
flux
log λ
400
350
300
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

```
#  16
```

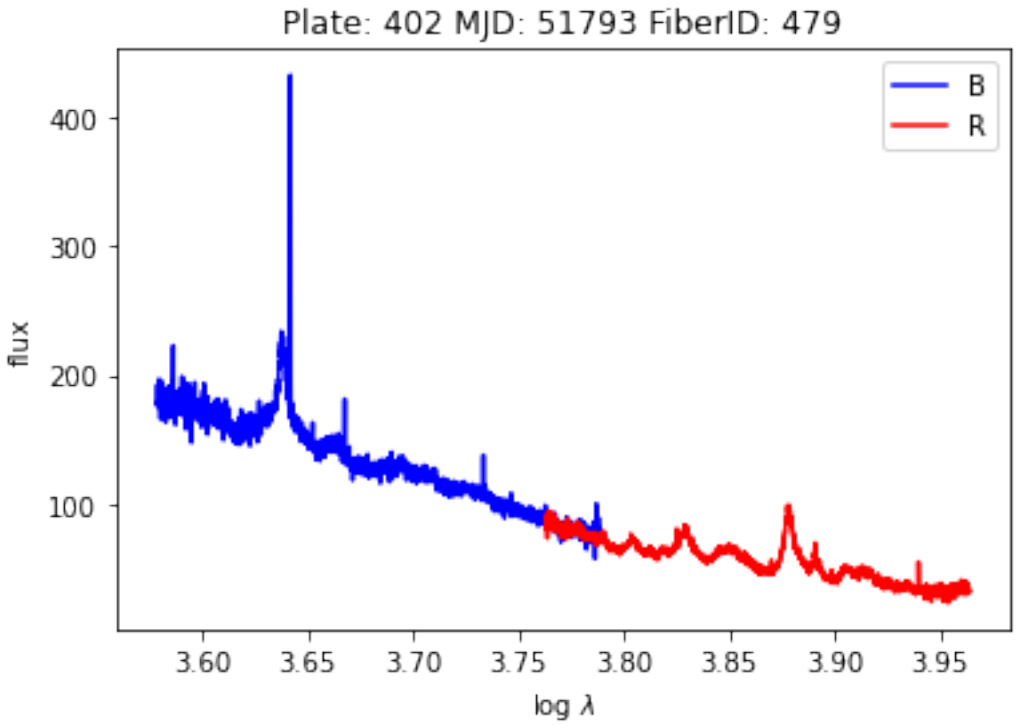


Plate: 402 MJD: 51793 FiberID: 479
B
R
400
300
200
100
flux
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

```
#  17
```

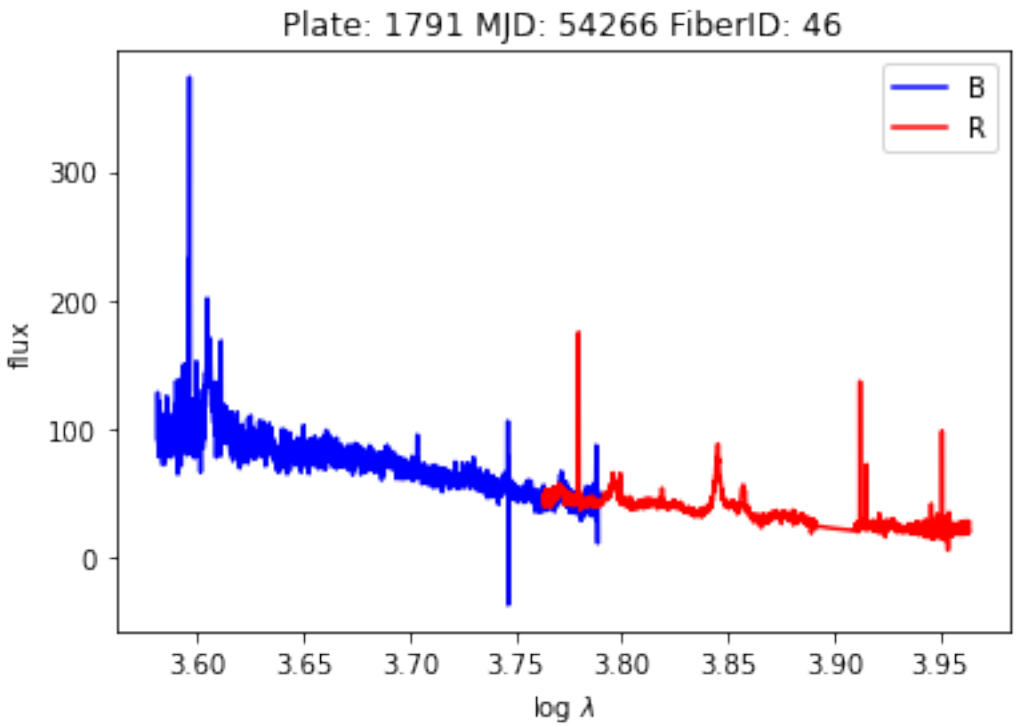


Plate: 1791 MJD: 54266 FiberID: 46
B
R
flux
300
200
100
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

```
#  18
```

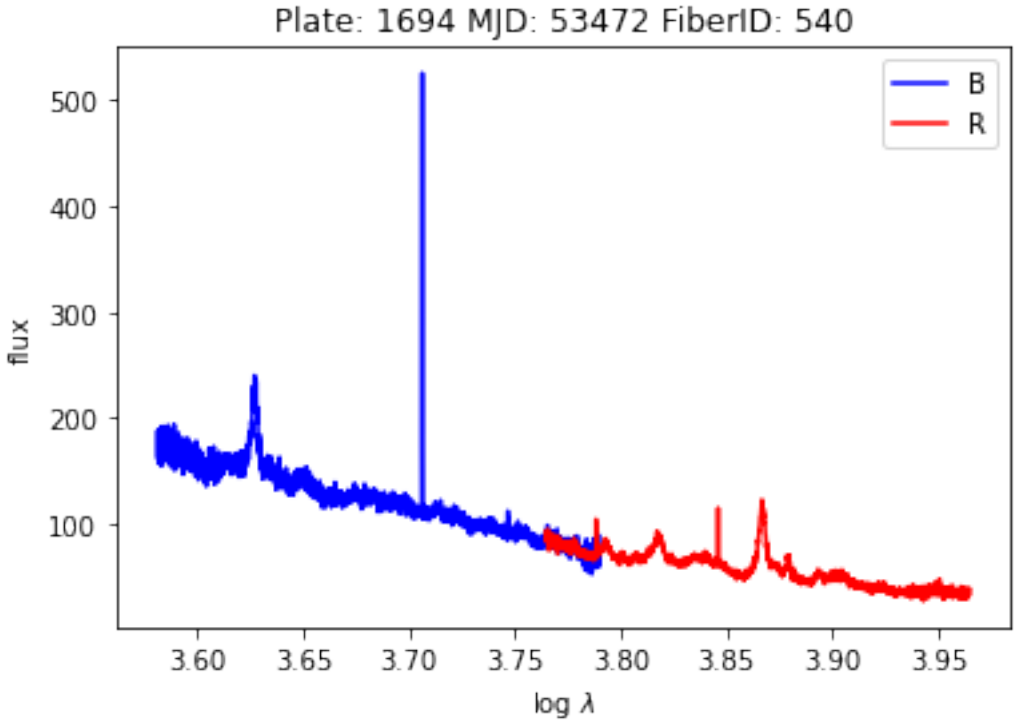


Plate: 1694 MJD: 53472 FiberID: 540
B
R
flux
500
400
300
200
100
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

19

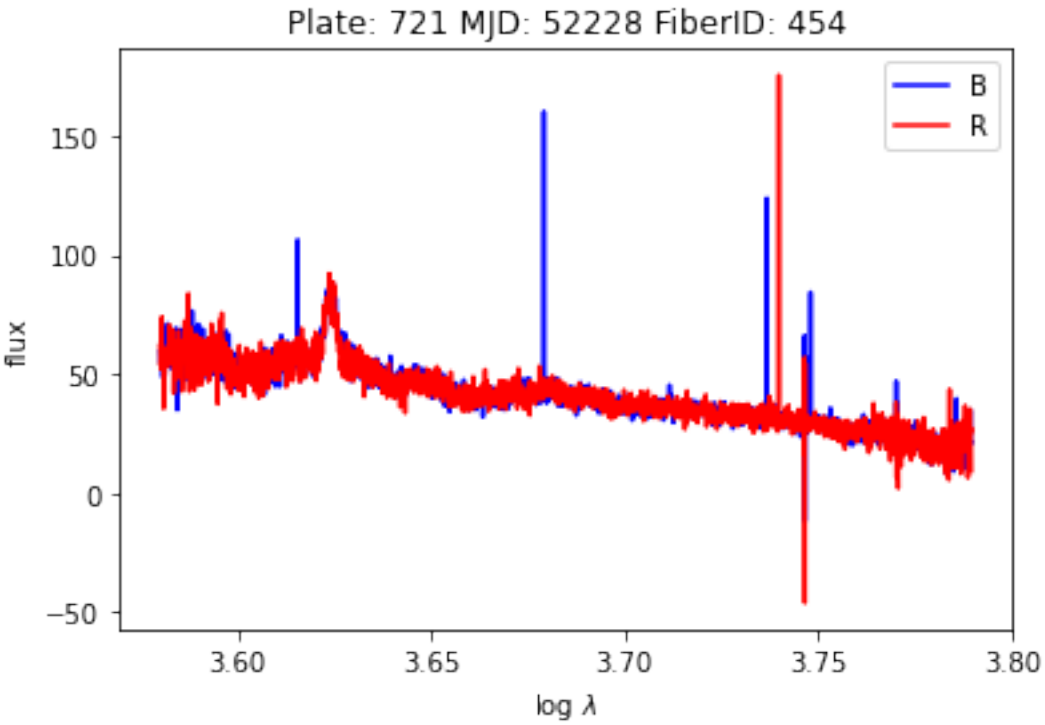


Plate: 721 MJD: 52228 FiberID: 454
B
R
150
100
50
0
−50
flux
3.60
3.65
3.70
3.75
3.80
log λ

20

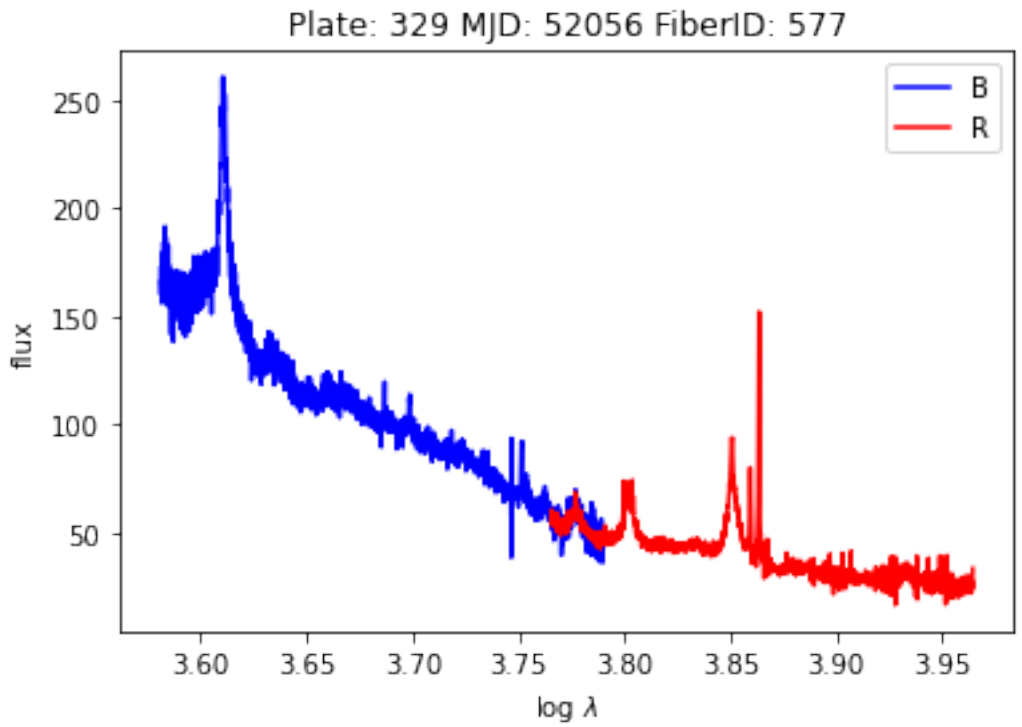


Plate: 329 MJD: 52056 FiberID: 577
B
R
flux
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

```
#  21
```

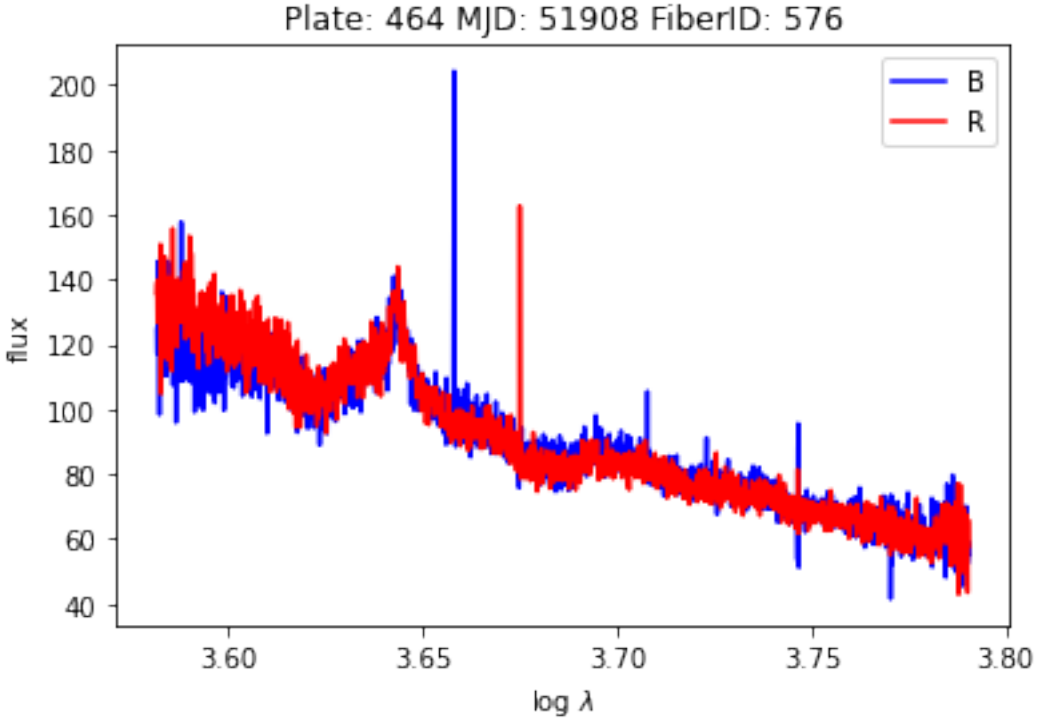


Plate: 464 MJD: 51908 FiberID: 576
B
R
flux
200
180
160
140
120
100
80
60
40
3.60
3.65
3.70
3.75
3.80
log λ

22

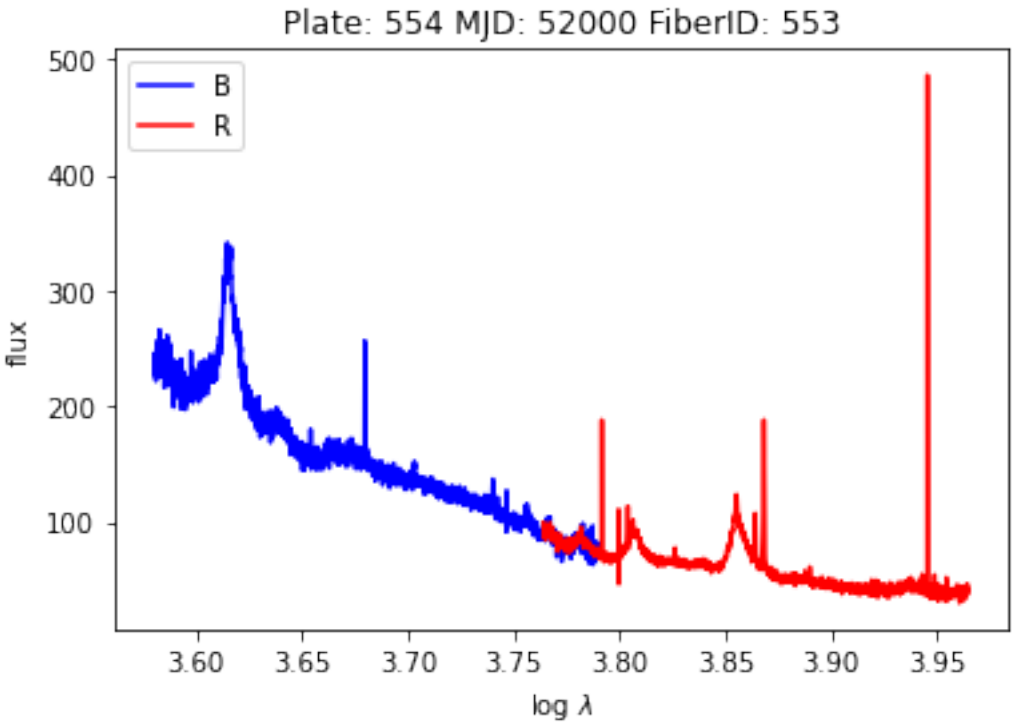


Plate: 554 MJD: 52000 FiberID: 553
B
R
flux
500
400
300
200
100
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

23

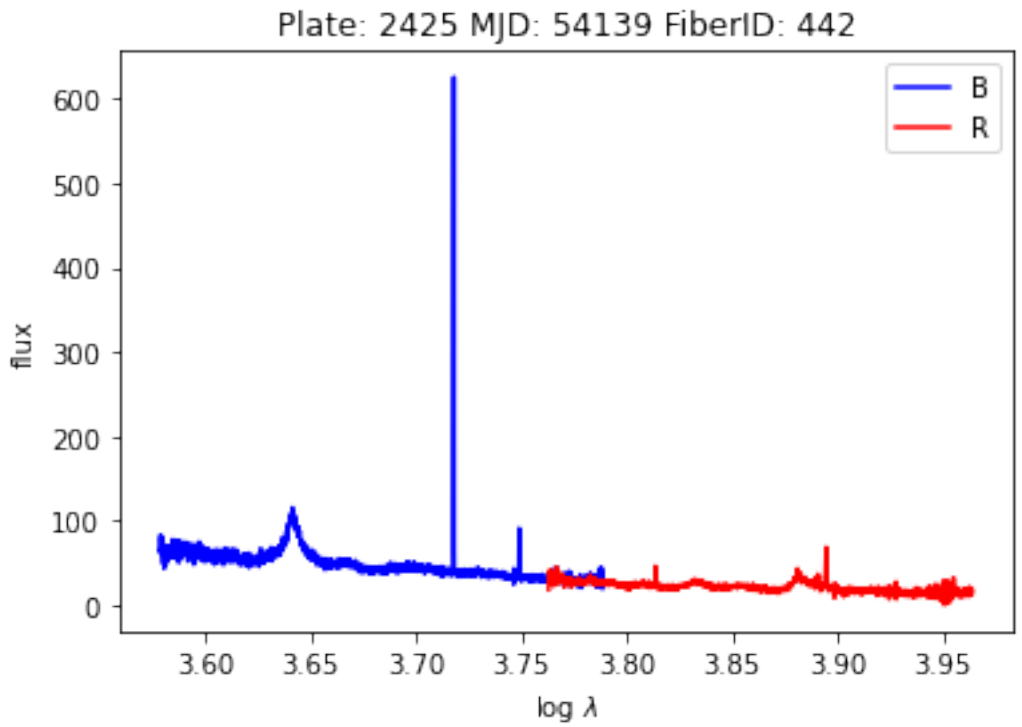


Plate: 2425 MJD: 54139 FiberID: 442
B
R
flux
log λ
0
100
200
300
400
500
600
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

24

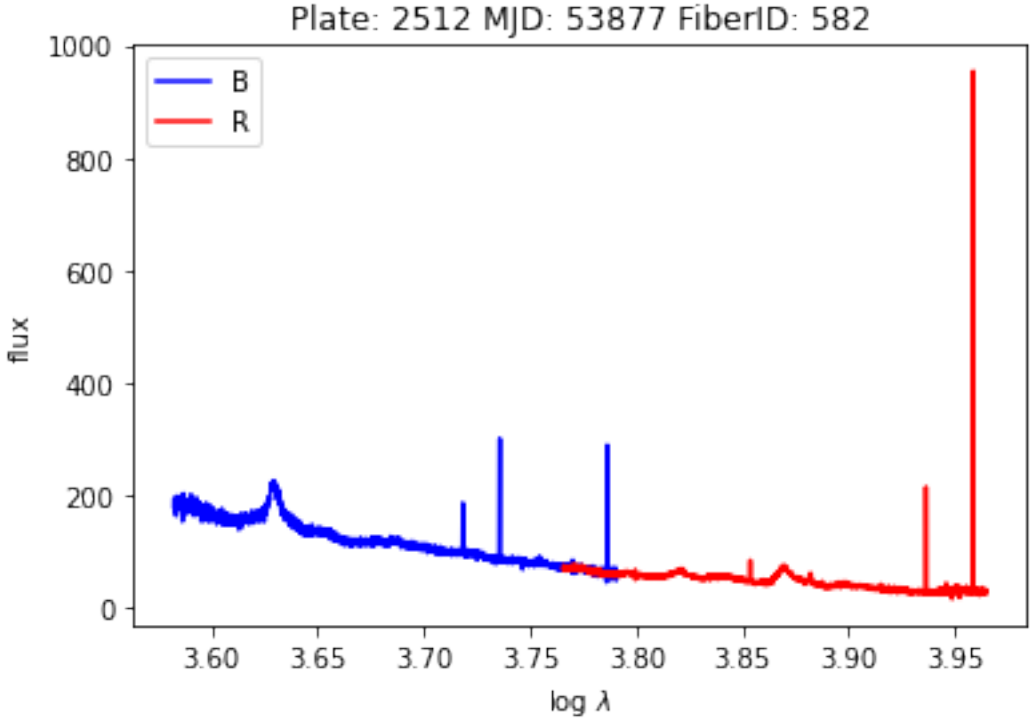


Plate: 2512 MJD: 53877 FiberID: 582
B
R
flux
log λ
0
200
400
600
800
1000
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

25

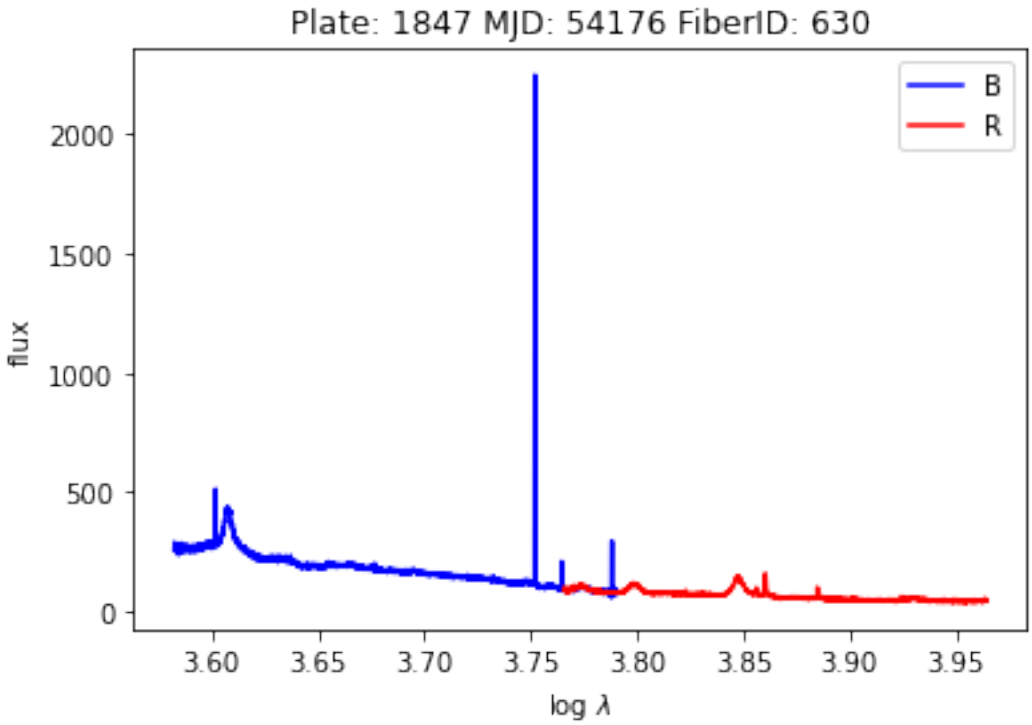


Plate: 1847 MJD: 54176 FiberID: 630
B
R
flux
2000
1500
1000
500
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

26

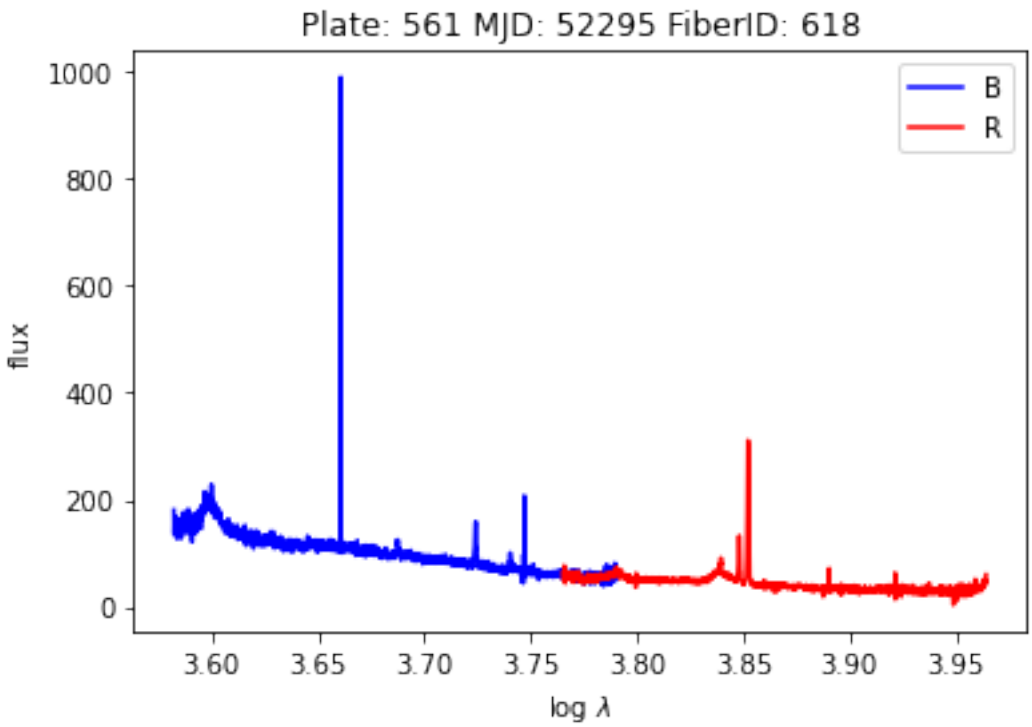


Plate: 561 MJD: 52295 FiberID: 618
B
R
flux
log λ
1000
800
600
400
200
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

27

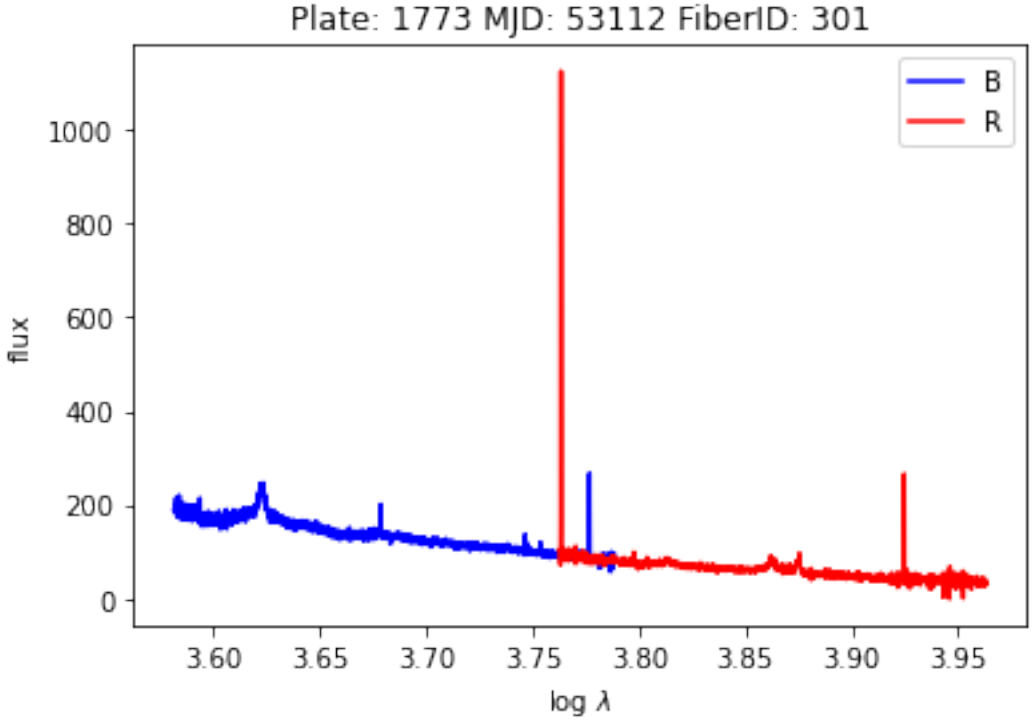


Plate: 1773 MJD: 53112 FiberID: 301
B
R
flux
log λ
1000
800
600
400
200
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

28

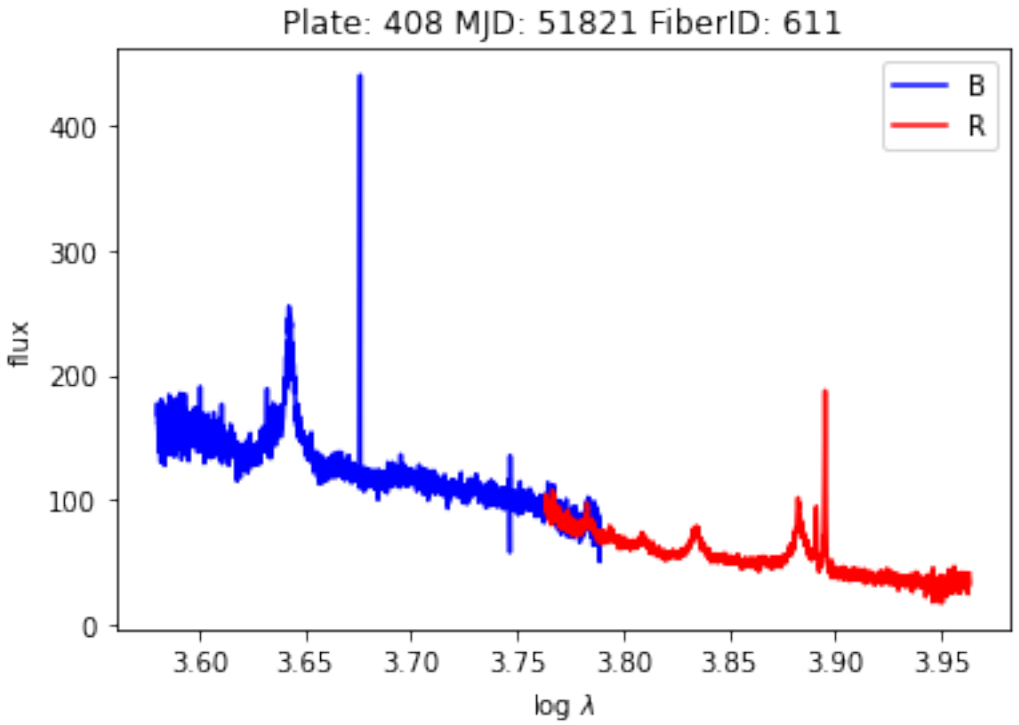


Plate: 408 MJD: 51821 FiberID: 611
B
R
flux
log λ
0
100
200
300
400
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

29

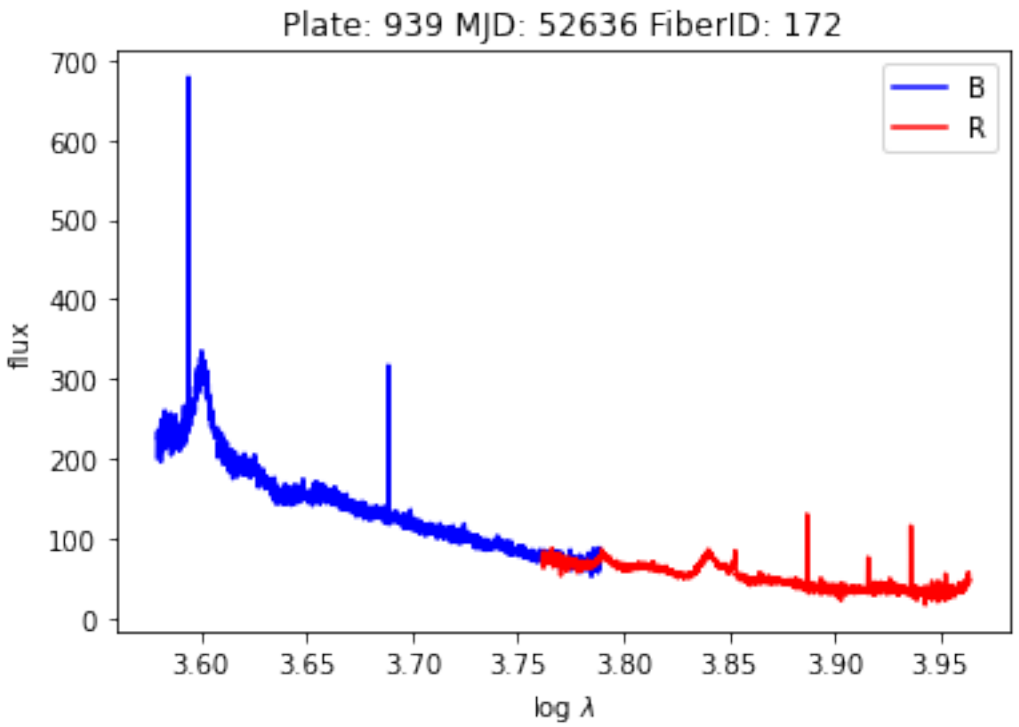


Plate: 939 MJD: 52636 FiberID: 172
B
R
flux
log λ
0
100
200
300
400
500
600
700
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

30

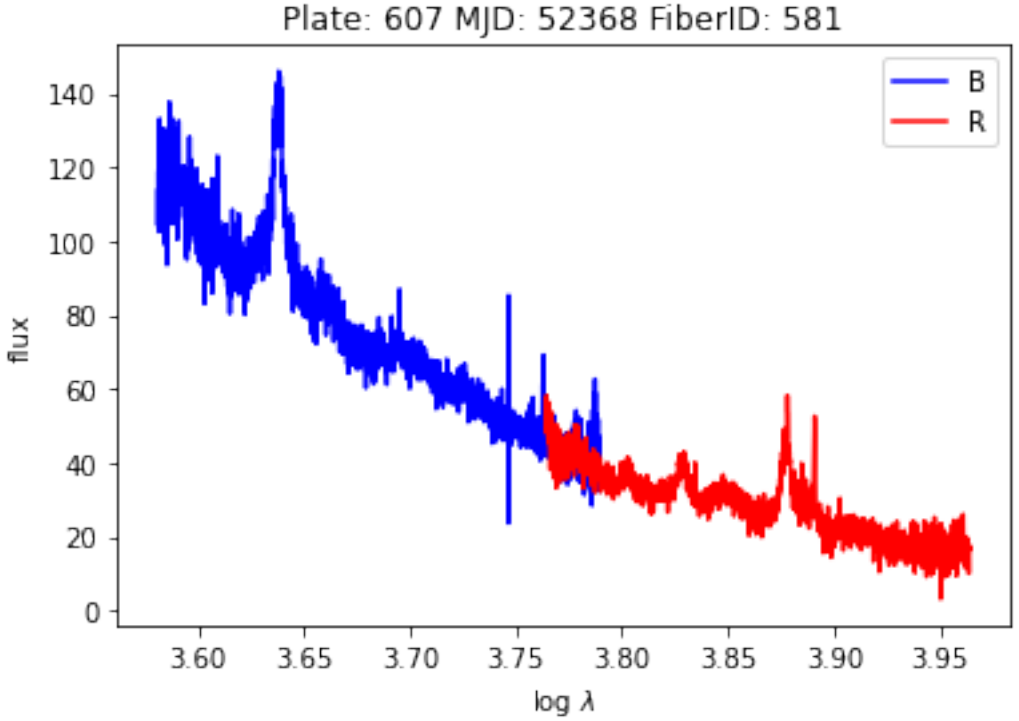


Plate: 607 MJD: 52368 FiberID: 581
B
R
flux
log λ
0
20
40
60
80
100
120
140
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

```
#  31
```

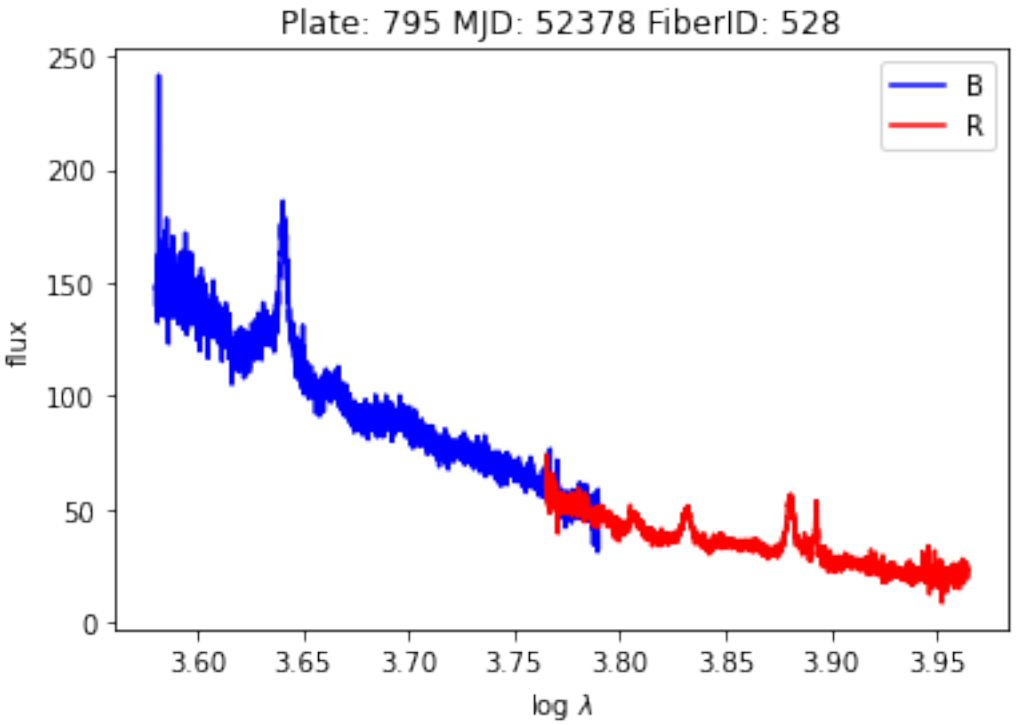


Plate: 795 MJD: 52378 FiberID: 528
B
R
flux
log λ
250
200
150
100
50
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

```
#  32
```

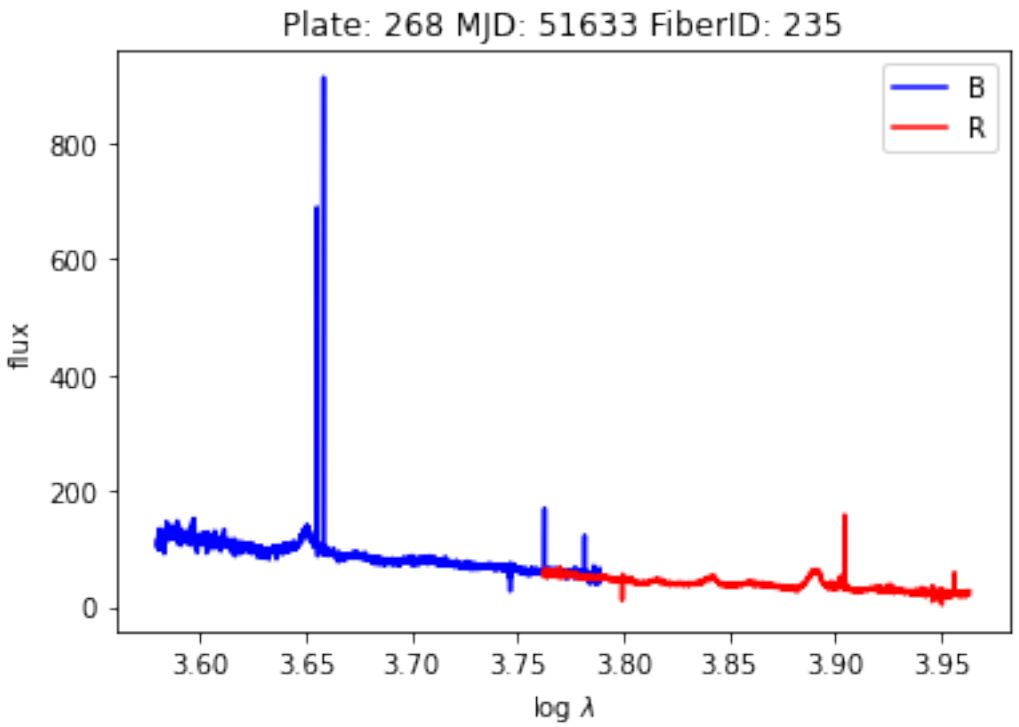


Plate: 268 MJD: 51633 FiberID: 235
B
R
flux
800
600
400
200
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

33

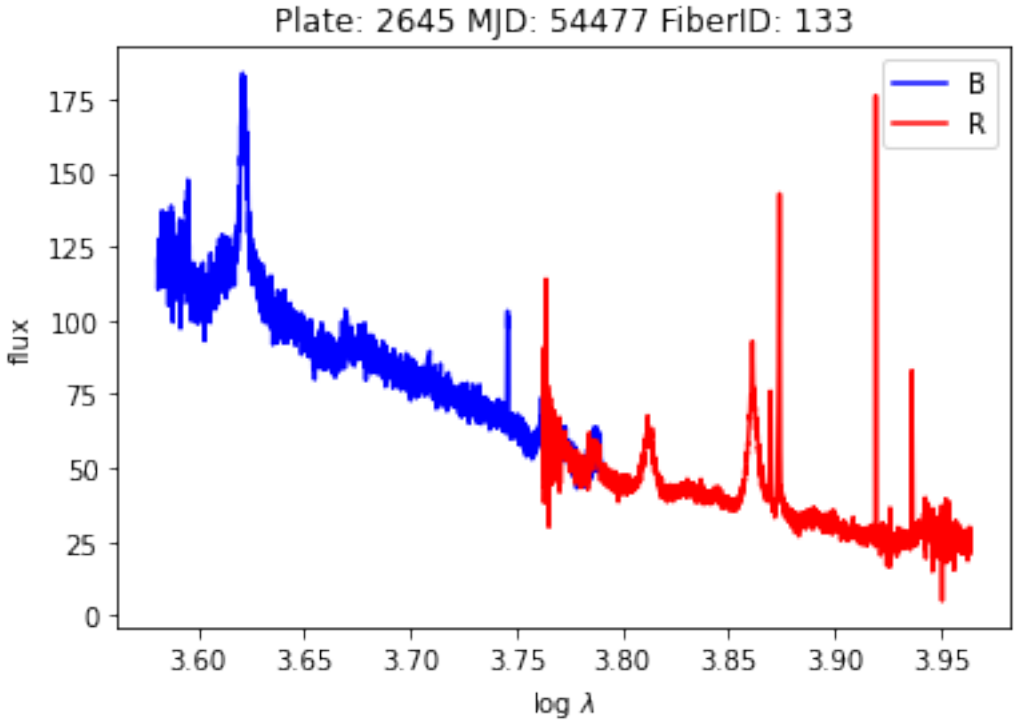


Plate: 2645 MJD: 54477 FiberID: 133
B
R
flux
175
150
125
100
75
50
25
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

34

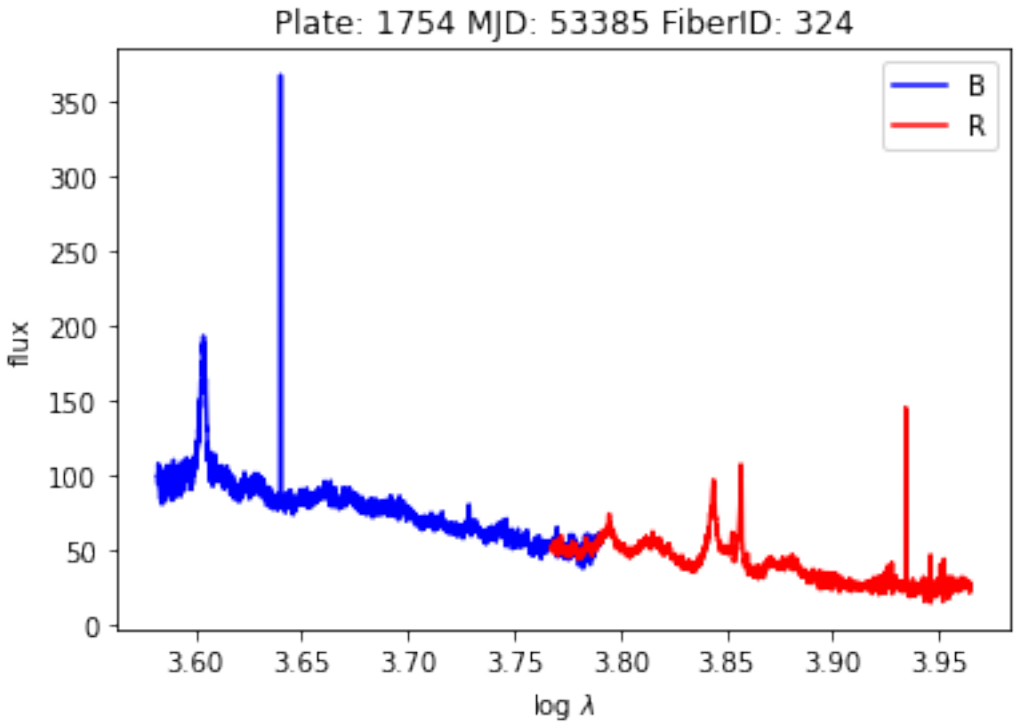


Plate: 1754 MJD: 53385 FiberID: 324
B
R
flux
log λ
0
50
100
150
200
250
300
350
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

35

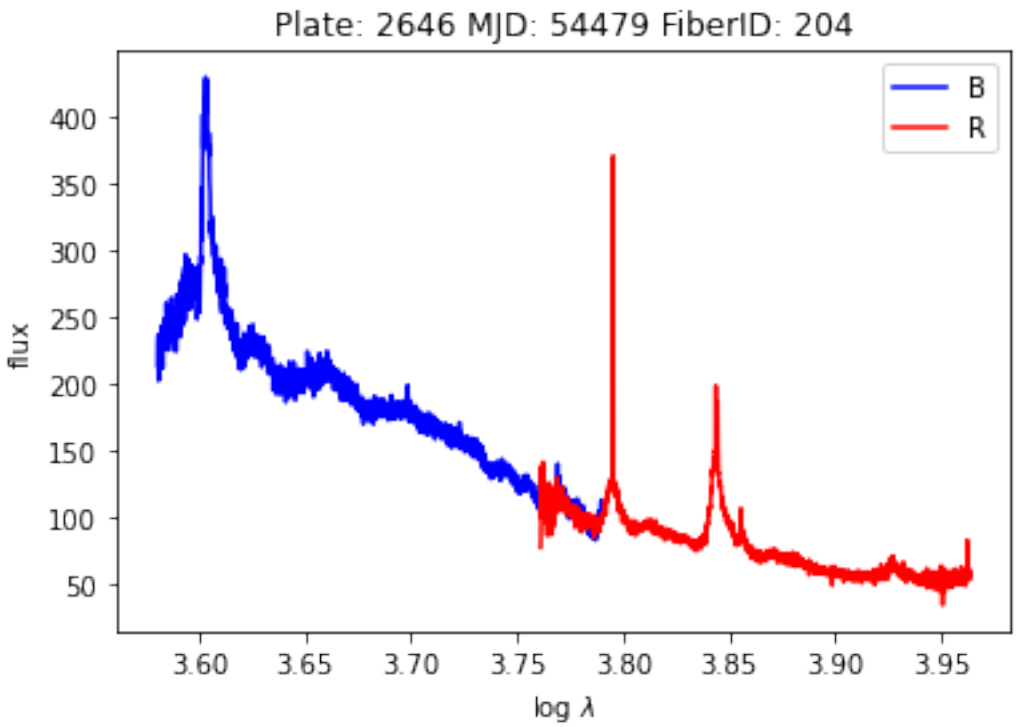


Plate: 2646 MJD: 54479 FiberID: 204
B
R
flux
400
350
300
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

36

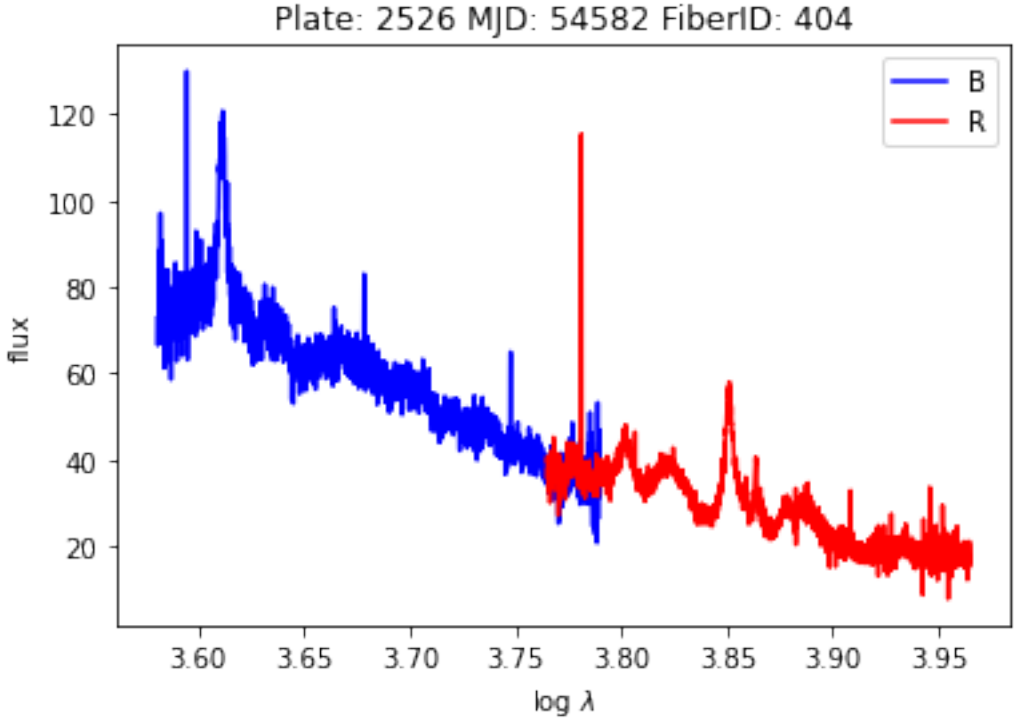


Plate: 2526 MJD: 54582 FiberID: 404
B
R
flux
120
100
80
60
40
20
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

37

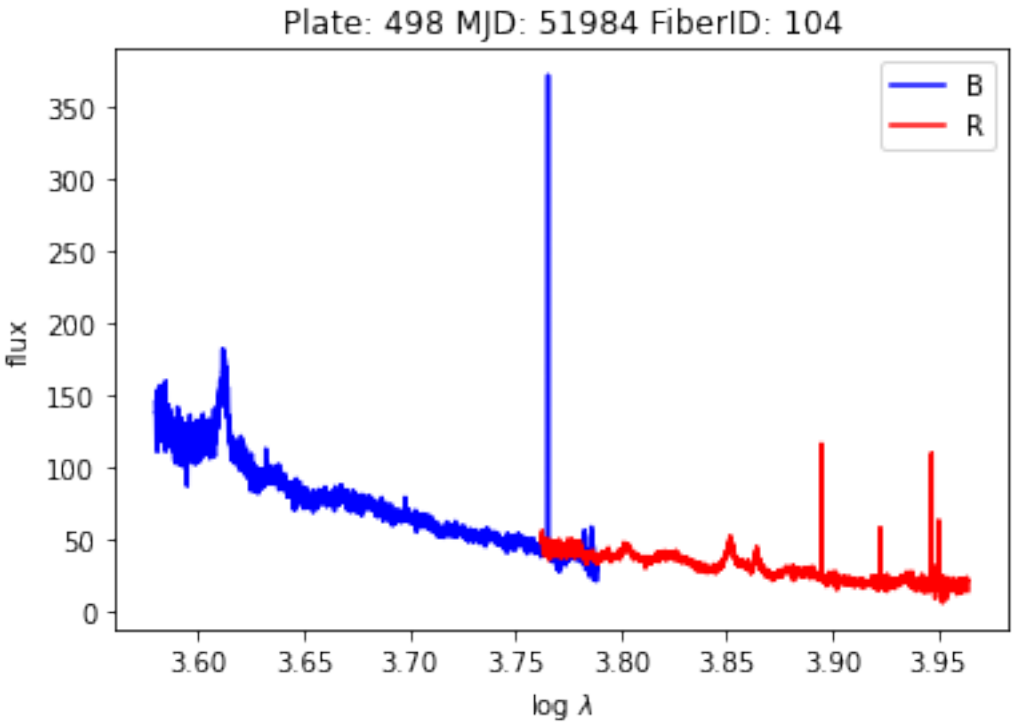


Plate: 498 MJD: 51984 FiberID: 104
B
R
flux
log λ
0
50
100
150
200
250
300
350
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

38

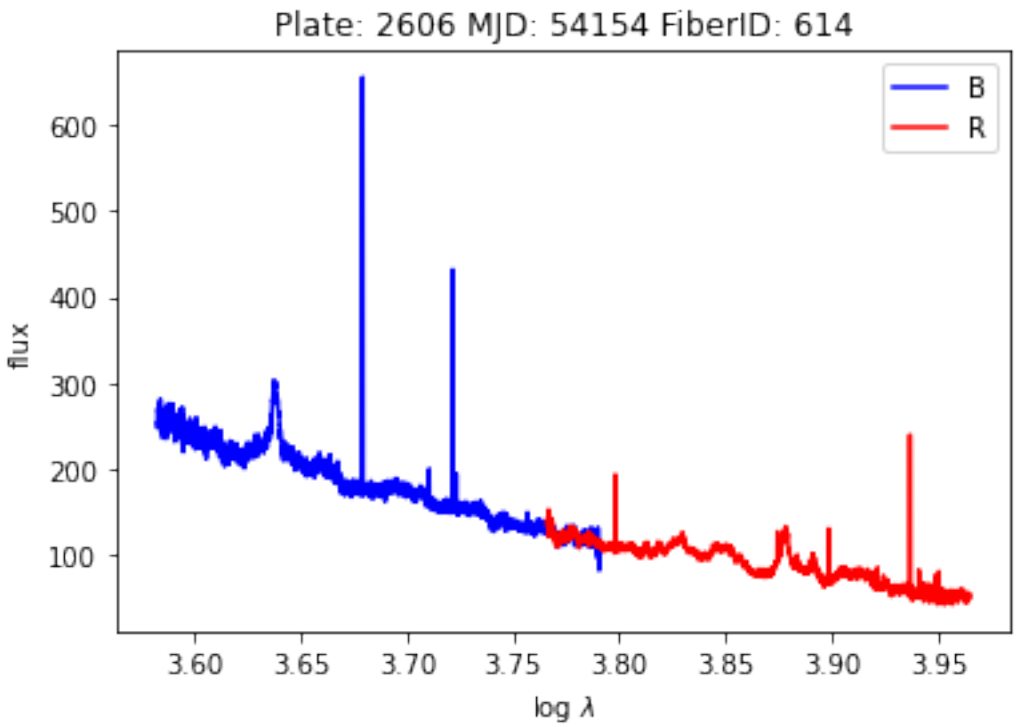


Plate: 2606 MJD: 54154 FiberID: 614
B
R
flux
600
500
400
300
200
100
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

39

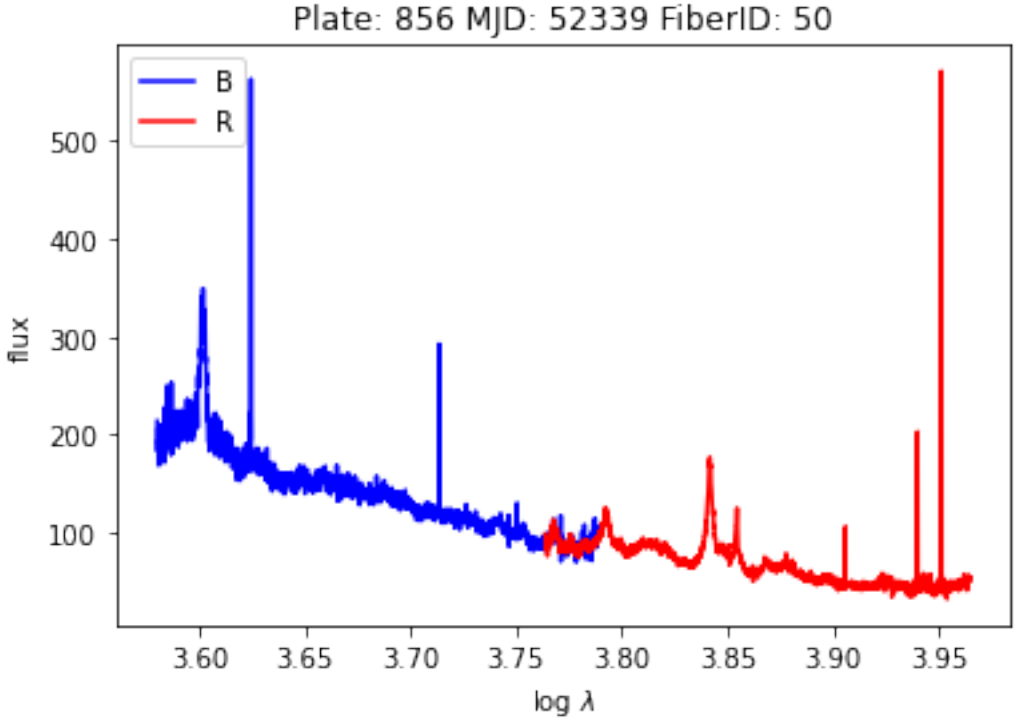


Plate: 856 MJD: 52339 FiberID: 50
B
R
flux
500
400
300
200
100
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

```
#  40
```

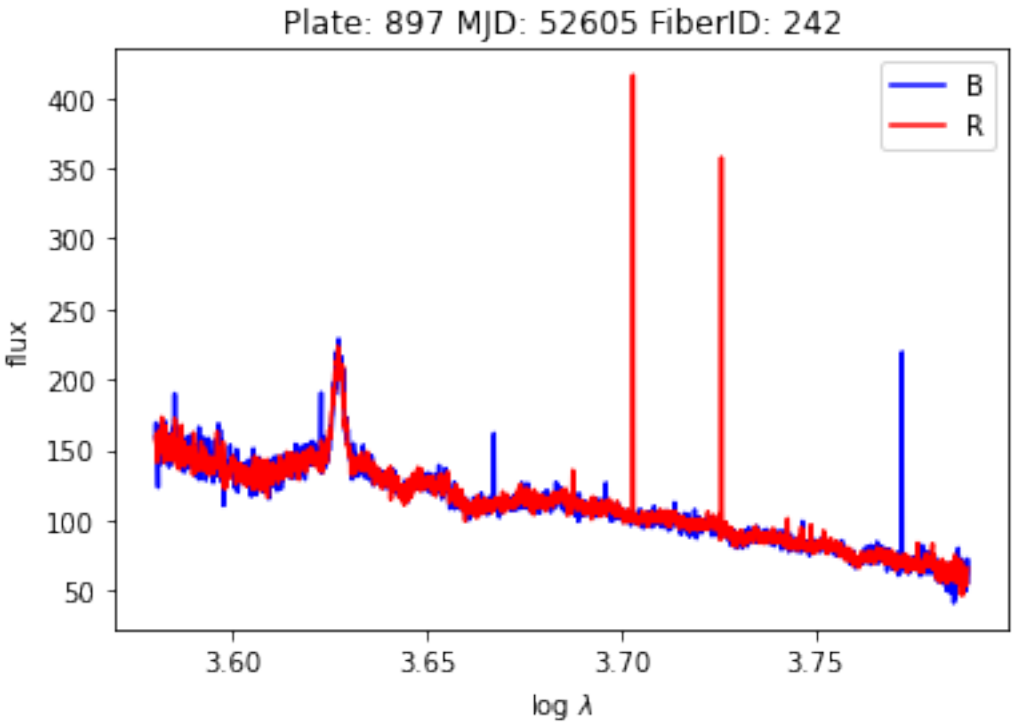


Plate: 897 MJD: 52605 FiberID: 242
B
R
flux
400
350
300
250
200
150
100
50
3.60
3.65
3.70
3.75
log λ

```
#  41
```

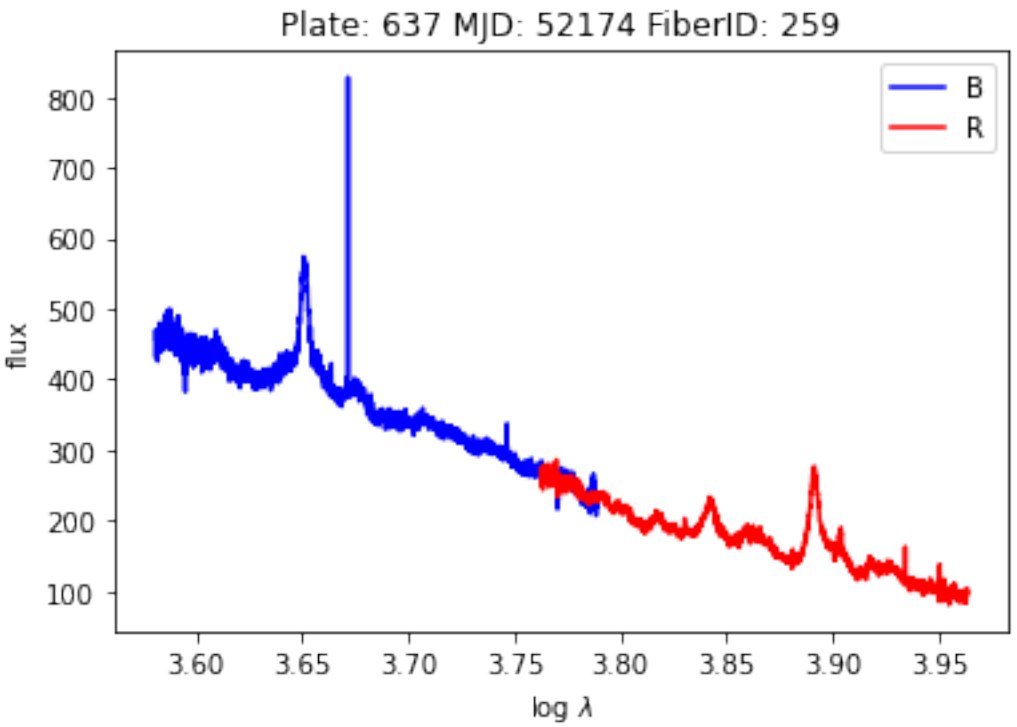


Plate: 637 MJD: 52174 FiberID: 259
B
R
flux
log λ
100
200
300
400
500
600
700
800
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95

42

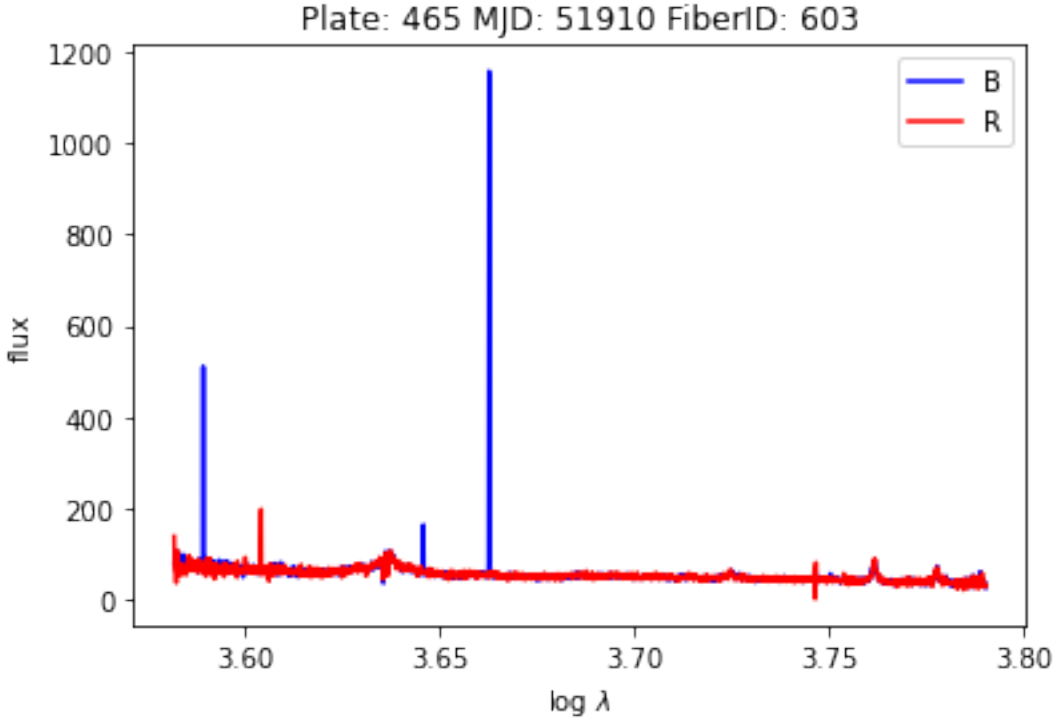


Plate: 465 MJD: 51910 FiberID: 603
B
R
flux
log λ
0
200
400
600
800
1000
1200
3.60
3.65
3.70
3.75
3.80

43

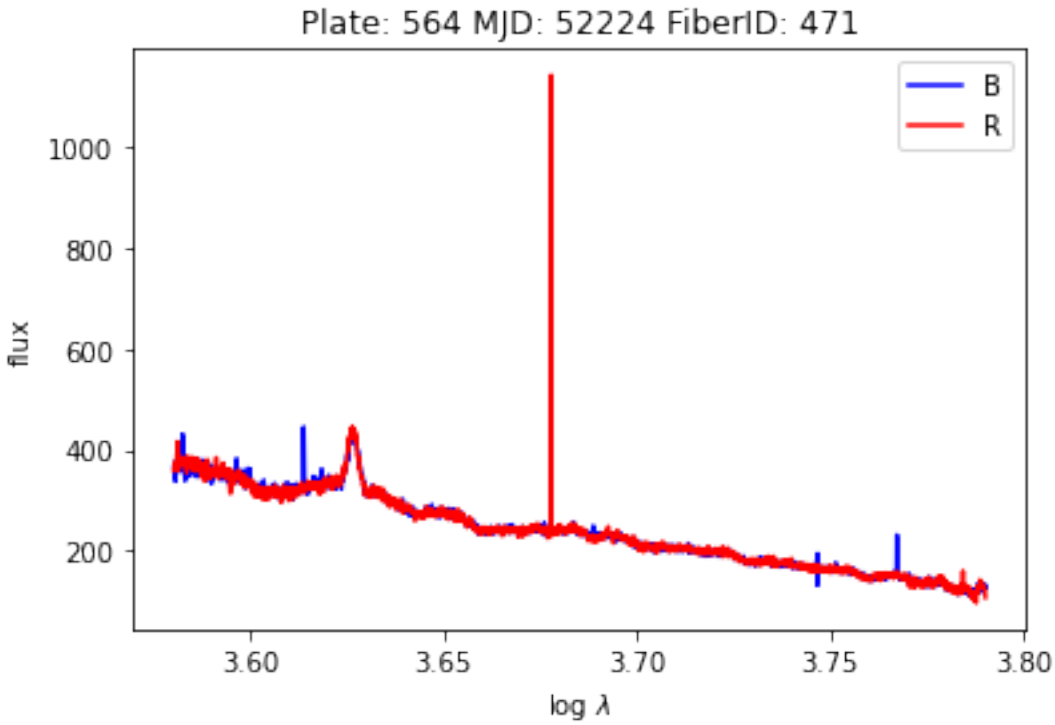


Plate: 564 MJD: 52224 FiberID: 471
B
R
1000
800
600
400
200
flux
3.60
3.65
3.70
3.75
3.80
log λ

44

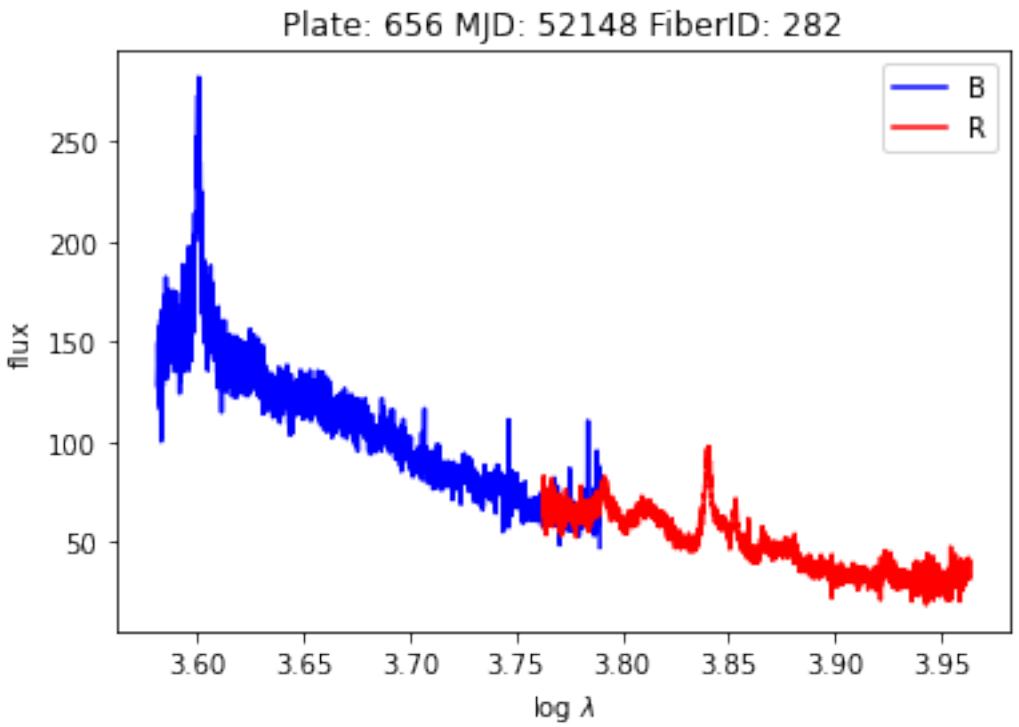


Plate: 656 MJD: 52148 FiberID: 282
B
R
flux
250
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

45

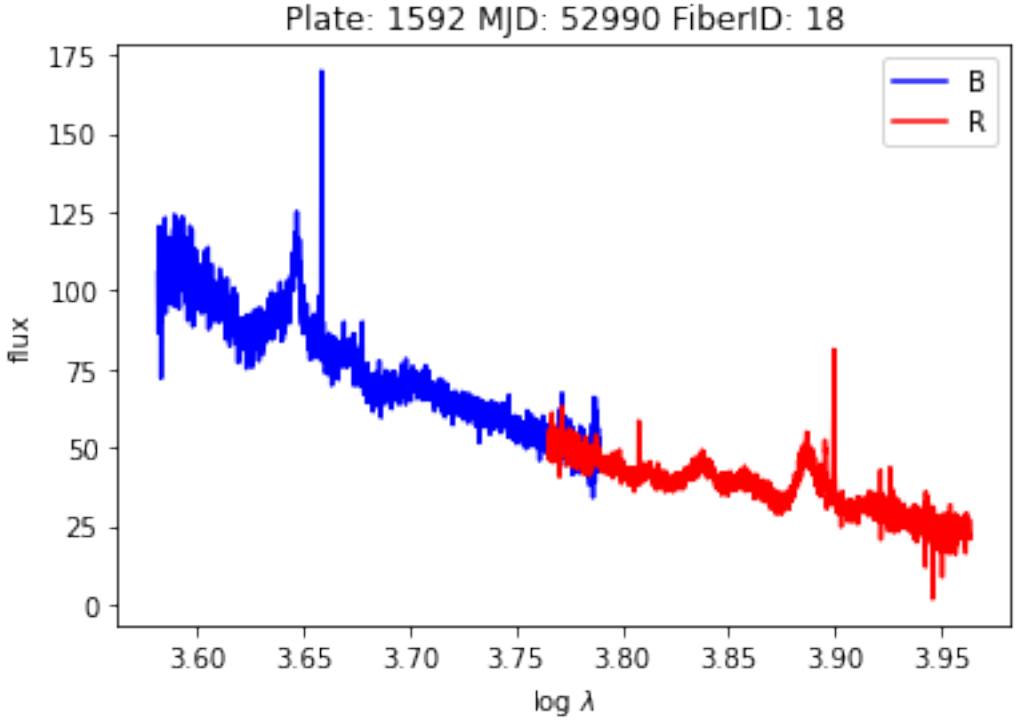


Plate: 1592 MJD: 52990 FiberID: 18
B
R
flux
175
150
125
100
75
50
25
0
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

46

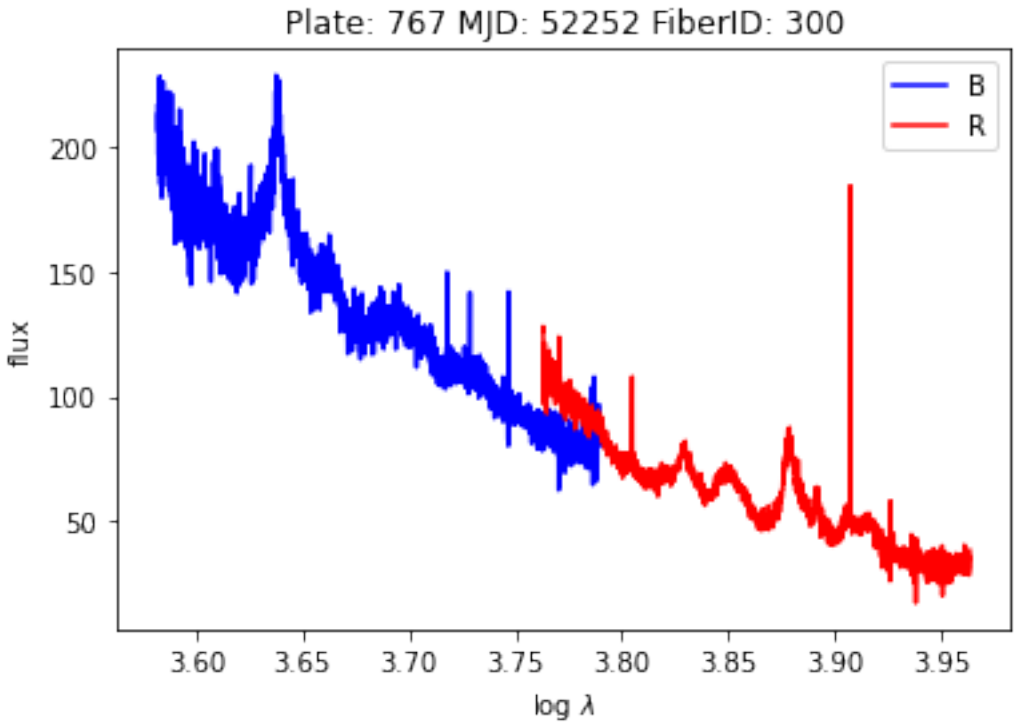
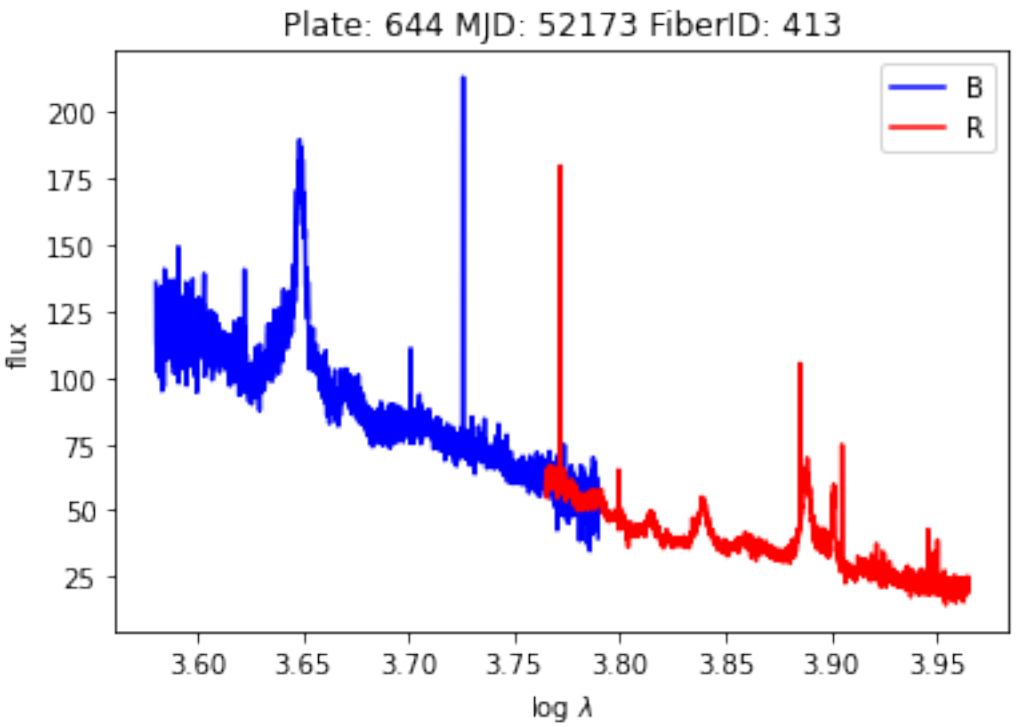


Plate: 767 MJD: 52252 FiberID: 300
B
R
flux
200
150
100
50
3.60
3.65
3.70
3.75
3.80
3.85
3.90
3.95
log λ

47



Note: From the results, we see 6 and 9 is not always the reduced result.