Analizando_datos

November 23, 2023

[2]:
```
# Import Python libraries to work with SciServer
import SciServer.CasJobs as CasJobs # query with CasJobs
import SciServer.SciDrive as SciDrive   # read/write to/from SciDrive
import SciServer.SkyServer as SkyServer   # show individual objects and␣
↪generate thumbnail images through SkyServer
print('SciServer libraries imported')
```

```
SciServer libraries imported
```

Importando las librerias necesarias para porder trabajar con la base de datos.

[7]:
```
# Import other libraries for use in this notebook.
import numpy as np                    # standard Python lib for math ops
#from scipy.misc import imsave         # save images as files
import pandas                         # data manipulation package
import matplotlib.pyplot as plt       # another graphing package
import os                             # manage local files in your Compute␣
↪containers
print('Supporting libraries imported')
```

```
Supporting libraries imported
```

Importando las liberias para trabajar en python.

[4]:
```
#import astroML
#from astroML.datasets import fetch_sdss_spectrum
from astropy.io import ascii
# Apply some special settings to the imported libraries
# ensure columns get written completely in notebook
pandas.set_option('display.max_colwidth', -1)
# do *not* show python warnings
import warnings
warnings.filterwarnings('ignore')
print('Settings applied')
```

```
Settings applied
```

```
<ipython-input-4-8d269ca52139>:6: FutureWarning: Passing a negative integer is
deprecated in version 1.0 and will not be supported in future version. Instead,
```

```
use None to not limit the column width.
  pandas.set_option('display.max_colwidth', -1)
```

Configuración para organizar los datos.

[5]:
```
# Find objects in the Sloan Digital Sky Survey's Data Release 14.
#
# Query the Sloan Digital Sky Serveys' Data Release 14.
# For the database schema and documentation see http://skyserver.sdss.org/dr14
#
# This query finds all galaxies with a size (petror90_r) greater than 10␣
 ↪arcseconds, within
# a region of sky with 100 < RA < 250, a redshift between 0.02 and 0.5, and a␣
 ↪g-band magnitude brighter than 17.
#
# First, store the query in an object called "query"
query="""
SELECT p.objId,p.ra,p.dec,p.petror90_r, p.expAB_r,
    p.dered_u as u, p.dered_g as g, p.dered_r as r, p.dered_i as i,
    s.z, s.plate, s.mjd, s.fiberid
FROM galaxy AS p
   JOIN SpecObj AS s ON s.bestobjid = p.objid
WHERE p.petror90_r > 10
  and p.ra between 100 and 250
  and s.z between 0.02 and 0.5
  and p.g < 17
"""
#Then, query the database. The answer is a table that is being returned to a␣
 ↪dataframe that we've named all_gals.
all_gals = CasJobs.executeQuery(query, "dr16")

print("SQL query finished.")
print("SQL query returned " + str(len(all_gals))+ " galaxies")
```

```
SQL query finished.
SQL query returned 40332 galaxies
```

Hacer la petición a en SQL

0.1 Galaxy colours

You will see in lectures that the optical colours of galaxies are related to the age of their stars - red galaxies hold older (redder, cooler) stars, whereas blue galaxies tend to have younger (bluer, hotted) stars. In practice, we can quantify "colour" in Astronomy as *the difference in magnitude in two different bands.*

The final exercises in the first notebook (SciServerLab_session1) give you a demonstration of how colours work in practice, and how the shape of the spectrum of a galaxy. We didn't consider it in the first notebook, but redshift can also affect the observed colour of galaxies (you will learn this in the later lectures, if you haven't yet).

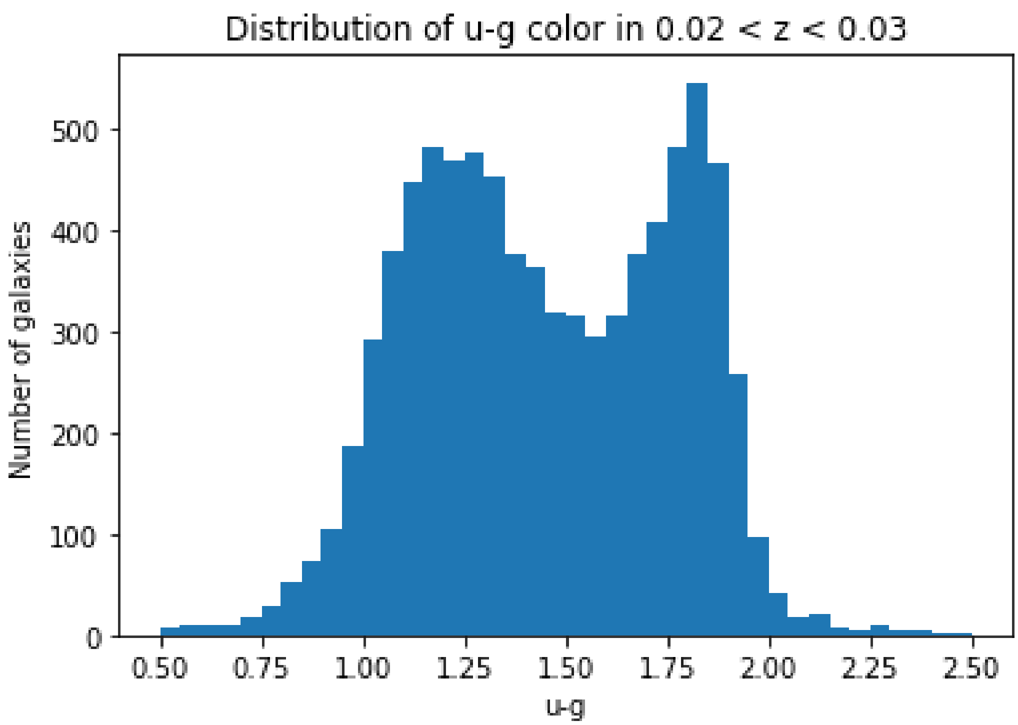
In this set of exercises **we will focus on the first slice in redshift**, which is very narrow, meaning that all galaxies have a similar redshift. I.e., if galaxies in this redshift slice have different colours, *it ought to be because their spectra and stellar composition are different*, and not because some are redshifted due to the expansion of the Universe.

The following cell plots a histogram of the values of the **u-g colour** of the galaxies in your dataframe:

[6]:
```
slice1 = np.where( (all_gals['z'] > 0.02) & (all_gals['z'] < 0.03))[0]

plt.hist(all_gals.loc[slice1]['u']-all_gals.loc[slice1]['g'], bins=40, range=(0.
↪5,2.5))
plt.xlabel('u-g')
plt.ylabel('Number of galaxies')
plt.title('Distribution of u-g color in 0.02 < z < 0.03')
```

[6]:
```
Text(0.5, 1.0, 'Distribution of u-g color in 0.02 < z < 0.03')
```



[9]:
```
median_umg = np.percentile(all_gals.loc[slice1]['u']-all_gals.loc[slice1]['g'],␣
↪50)
print(median_umg)
```

```
1.4351650000000005
```