Zach Warren

Homework 2

ASTR 3800

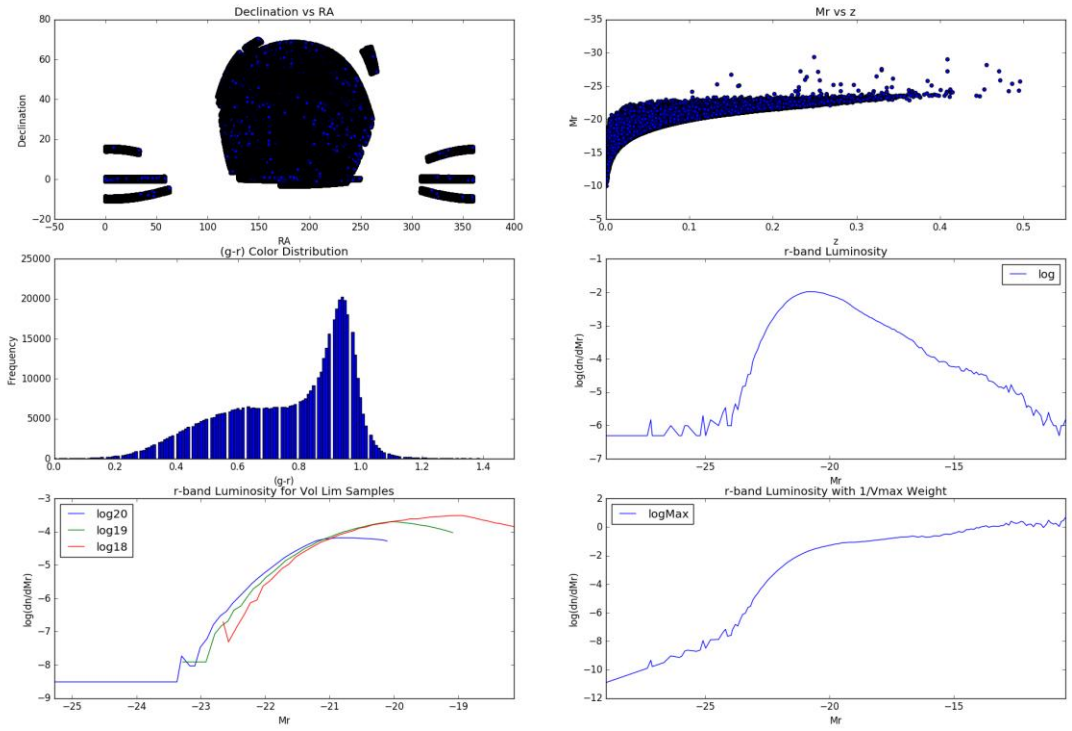
February 23, 2016

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.cbook as cbook
import math

Location = r'C:\Users\Zachary Warren\OneDrive\S2 2015-2016\ASTR
3800\Homework2\src\SDSS_DR7.dat'
df = pd.read_csv(Location, delimiter = ' ', names = ['RA', 'Declination', 'z', 'Mg',
'Mr'], skipinitialspace = True)

f, plts = plt.subplots(3,2,figsize=(23,15))

#part 1
df.plot(kind='scatter', x='RA',y='Declination', ax=plts[0,0])

df.plot(kind='scatter', x='z',y='Mr', ax=plts[0,1])
plts[0,1].invert_yaxis()
plts[0,1].set_xlim([0.0,.55])

#part 2
df['Mg-Mr'] = df['Mg'] - df['Mr']

df['Mg-Mr'].plot(kind='hist',ax=plts[1,0], bins=1200)

plts[1,0].set_xlim([0.0, 1.5])

#calculate fraction
def countNums(row):
    if row['Mg-Mr'] >= .75:
        return 1
    if row['Mg-Mr'] < .75:
        return 0

df['>.75'] = df.apply(lambda row: countNums(row), axis=1)

totalBlue = df['>.75'].sum()
totalRows = len(df.axes[0])
blueFraction = totalBlue/totalRows

print('Total blue: ', totalBlue)
print('Total galaxies:', totalRows)
print('Blue Fraction: ', blueFraction)

#part 3

def log(row,col,logVol):
    if row[col] == 0:
        return -logVol
    else:
        return np.log10(row[col]) -logVol

def rBand(df,col,bin,logR,logVal):
    bins = np.arange(df[col].min(),df[col].max(), .1)
```

```
    df[bin] = pd.cut(df[col], bins)
    group = df[col].groupby(df[bin])
    groupSize = group.size()
    gd = pd.DataFrame(groupSize)
    gd['magAv']=df[col].groupby(df[bin]).mean()
    gd=gd.reset_index()

    gd[logR]=gd.apply(lambda row: log(row,1,logVal),axis=1)

    gd=gd.dropna()

    return gd


gd=rBand(df,'Mr','binnedMr','log',6.31)
gd.plot(x='magAv', y='log', ax=plts[1,1])


#part 4
def findGalaxies(row,mag,z):
    if row['z'] < z and row['Mr'] < mag:
        return 1

def volume(z):
    d = (3*math.pow(10,3)*z)
    volume = (2.295/3.0)*math.pi*math.pow(d,3)
    return volume

def countBlues(row, sample):
    if (row[sample] ==1):
        if row['Mg-Mr'] >= .75:
            return 1
        if row['Mg-Mr'] < .75:
            return 0


#volume sample 1
#Mr <-20, z <.171
df['VSamp(-20)'] = df.apply(lambda row: findGalaxies(row,-20,.171), axis= 1)
total20 = df['VSamp(-20)'].sum()
df['VSamp(-20)Vals']=df['Mr'].loc[df['VSamp(-20)'] == 1]

df['>.75(-20)'] = df.apply(lambda row: countBlues(row, 'VSamp(-20)'), axis=1)
total20Blue = df['>.75(-20)'].sum()

vol20=volume(.171)
logVol20=np.log10(vol20)

d20=rBand(df,'VSamp(-20)Vals','binned20','log20',logVol20)
d20.plot(x='magAv',y='log20',ax=plts[2,0])

print('V Sample One:')
print('Redshift Bound: ', .171)
print('Total Galaxies: ', total20)
print('Total Volume: ', vol20, 'h-3 Mpc3')
print('Blue Galaxy Fraction: ', total20Blue/total20, '\n')

#volume sample 2
#Mr <-19, z < .108
df['VSamp(-19)'] = df.apply(lambda row: findGalaxies(row,-19,.108), axis= 1)
total19 = df['VSamp(-19)'].sum()
df['VSamp(-19)Vals']=df['Mr'].loc[df['VSamp(-19)'] == 1]
```

```
df['>.75(-19)'] = df.apply(lambda row: countBlues(row, 'VSamp(-19)'), axis=1)
total19Blue = df['>.75(-19)'].sum()

vol19=volume(.108)
logVol19=np.log10(vol19)

d19=rBand(df,'VSamp(-19)Vals','binned19','log19',logVol19)
d19.plot(x='magAv',y='log19',ax=plts[2,0],color='g')

print('V Sample Two:')
print('Redshift Bound: ', .108)
print('Total Galaxies: ', total19)
print('Total Volume: ', volume(.108), 'h-3 Mpc3')
print('Blue Galaxy Fraction: ', total19Blue/total19, '\n')

#volume sample 3
#Mr < -18 z <.068
df['VSamp(-18)'] = df.apply(lambda row: findGalaxies(row,-18,.068), axis= 1)
total18 = df['VSamp(-18)'].sum()
df['VSamp(-18)Vals']=df['Mr'].loc[df['VSamp(-18)'] == 1]

df['>.75(-18)'] = df.apply(lambda row: countBlues(row, 'VSamp(-18)'), axis=1)
total18Blue = df['>.75(-18)'].sum()

vol18=volume(.068)
logVol18=np.log10(vol18)

d18=rBand(df,'VSamp(-18)Vals','binned18','log18',logVol18)
d18.plot(x='magAv',y='log18',ax=plts[2,0], color='r')

print('V Sample Three:')
print('Redshift Bound: ', .068)
print('Total Galaxies: ', total18)
print('Total Volume: ', volume(.068), 'h-3 Mpc3')
print('Blue Galaxy Fraction: ', total18Blue/total18, '\n')


#part 5
def z(mag):
    if mag == None:
        return .1
    else:
        return math.pow(10,((mag-17.77)/(-5) -9.322))

def log2(row,col,z):
    if row[col] == 0:
        return -np.log10(volume(z))
    else:
        return np.log10(row[col]) -np.log10(volume(z))

def rBand2(df,col,bin,logR):
    bins = np.arange(df[col].min(),df[col].max(), .1)
    df[bin] = pd.cut(df[col], bins)
    group = df[col].groupby(df[bin])
    groupSize = group.size()
    gd = pd.DataFrame(groupSize)
    gd['magAv']=df[col].groupby(df[bin]).mean()
    gd=gd.reset_index()

    gd['MagMr'] = gd.apply(lambda row: z(row['magAv']), axis=1)
    gd[logR]=gd.apply(lambda row: log2(row,1,row['MagMr']),axis=1)

    gd=gd.dropna()
```

```
    return gd

gMax=rBand2(df,'Mr','binnedMr','logMax')
gMax.plot(x='magAv', y='logMax', ax=plts[2,1])



#Titles and Axes labels
plts[0,0].set_title('Declination vs RA')

plts[0,1].set_title('Mr vs z')

plts[1,0].set_xlabel('(g-r)')
plts[1,0].set_title('(g-r) Color Distribution')

plts[1,1].set_xlabel('Mr')
plts[1,1].set_ylabel('log(dn/dMr)')
plts[1,1].set_title('r-band Luminosity')

plts[2,0].set_xlabel('Mr')
plts[2,0].set_ylabel('log(dn/dMr)')
plts[2,0].set_title('r-band Luminosity for Vol Lim Samples')

plts[2,1].set_xlabel('Mr')
plts[2,1].set_ylabel('log(dn/dMr)')
plts[2,1].set_title('r-band Luminosity with 1/Vmax Weight')

plt.show()
```



Output from program:

Total blue: 328989

Total galaxies: 550166

Blue Fraction: 0.5979813365420619

V Sample One:

Redshift Bound: 0.171

Total Galaxies: 307242.0

Total Volume: 324461673.0042967 h-3 Mpc3

Blue Galaxy Fraction: 0.6774008761822928

V Sample Two:

Redshift Bound: 0.108

Total Galaxies: 257626.0

Total Volume: 81742203.0837476 h-3 Mpc3

Blue Galaxy Fraction: 0.590258747176139

V Sample Three:

Redshift Bound: 0.068

Total Galaxies: 128634.0

Total Volume: 20403365.531192 h-3 Mpc3

Blue Galaxy Fraction: 0.4873128410840058

The luminosity functions made from all of the volume limited samples show more accurately the amount of galaxies at lower luminosities. The flux limited sample shows more of the very luminous galaxies because we can see more of them, even though we know there should be more less luminous galaxies. The weighted sample does the same thing by adjusting the 'amount' of galaxies at far distances (the more luminous ones) and increasing the 'amount' of galaxies nearby, essentially increasing the number of low luminosity galaxies vs high luminosity galaxies.