Computational Physics Assignment 5

Dara Corr - 18483836 - 17/02/21

In this assignment we are tasked with analysing a data sheet from a text file by importing the file into python and saving its contents in arrays. Once this is done further data analysis and plots are possible.

In part 1 we open a file called StarData.txt containing 5 columns of data from Astronomers' observations of several thousand stars. The 5 columns are Star ID number, apparent V magnitude m_V, observed B-V colour ($m_B - m_V$), observed parallax in units of arcseconds and uncertainty in parallax in units of arcseconds. All of the Data is loaded and is entered into an array with each row as each entry in the array. Then a for loop takes each line and assigns each column entry to a value in its corresponding array i.e. Star ID number. Now there will be 5 arrays containing all the data from the 5 columns in the text file. Now we can use the data for further data analysis and plotting.

In the second part of the assignment, we plot the apparent magnitude of the stars vs colour on a scatter plot. This is a scatter plot has m_V plotted on the y-axis against $m_B - m_V$. This plot is caled a colour-magnitude diagram.

For Task 3, we are asked to plot a similar scatter plot as in task 2. Instead of using apparent magnitude here, we use absolute mangnitude M which corrects for the effects of distance on the brightness of a star. The absolute magnitude of a star is defined as the apparent magnitude of a star if viewed from a distance of 10 parsecs away. If the apparent magnitude and the distance to the star are known then we can find the absolute magnitude as follows:

$$m - M = 5log\left(\frac{d}{10}\right)$$

Note that the distance is found using the parallax angle p in the expression $d = \frac{1}{p}$. Plotting M against $B - V$ here gives us a plot which resembles the Hertzprung-Russel Diagram. Using a few additional lines of code we can count how many Main Sequence, Red Giant and White Dwarf stars are present on the diagram that we produced.

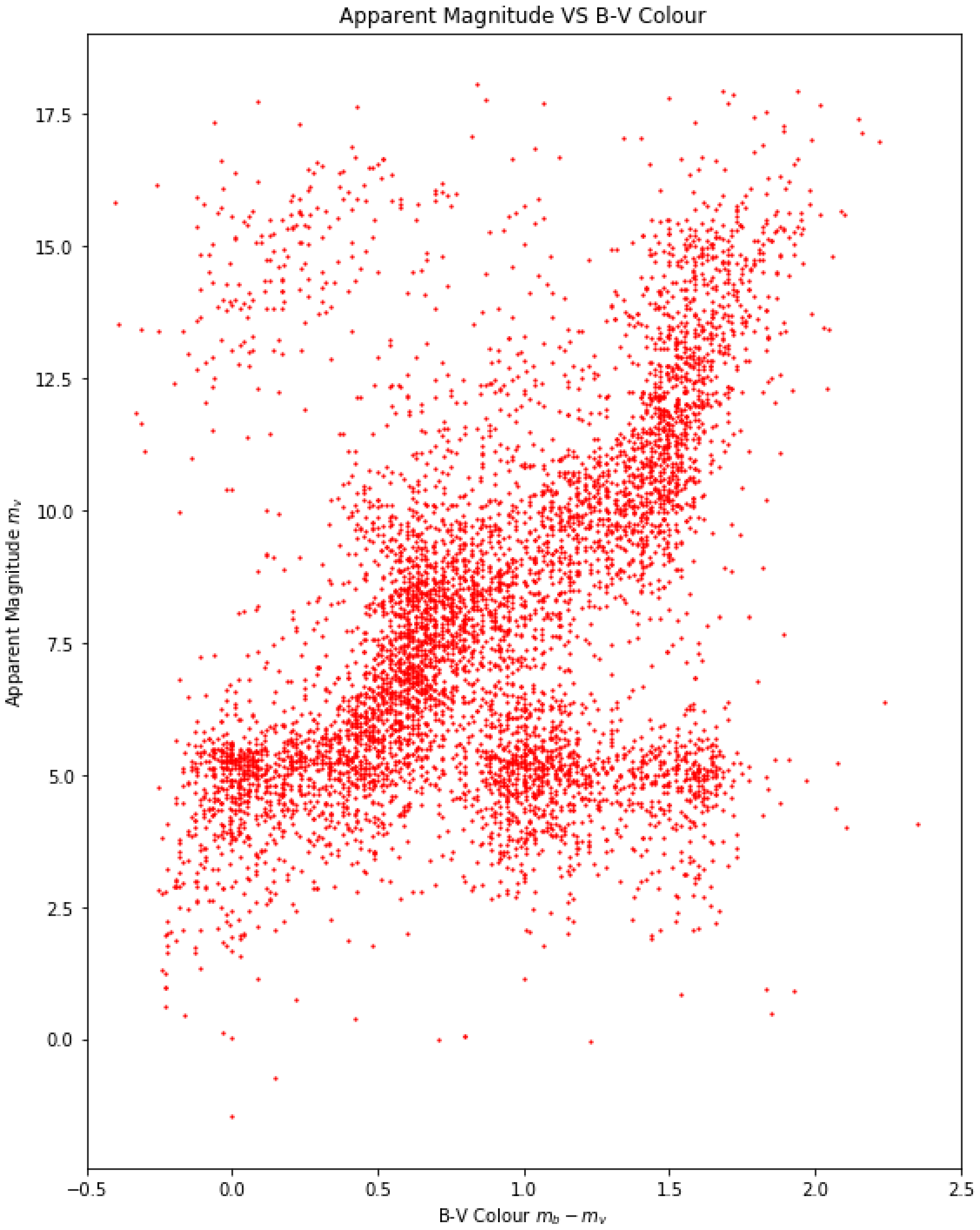
Additionally, a luminosity scale in terms of solar luminosities can be added as another vertical axis on the right hand of the diagram and colours on the plot can be changed to match the colours of the corresponding stars.

In [2]:

```
#import numpy and matplotlib
import numpy as np
import matplotlib.pyplot as plt


#task1

FileData = open('StarData.txt','r') #open file in read mode and denotes it by 'Filedata'

with open('StarData.txt') as StarData: #split each string of text from the file into a list
    lines = StarData.read().splitlines() #splitting is done at line breaks

#set up arrays to store each column of data in
StarID = []
V_Mag = []
B_V = []
Parallaxes = []
Parallax_Error = []

for line in lines: #line by line append data from every column to each corresponding array
    Columns = line.split()
    StarID.append(float(Columns[0])) #files are imported as strings so they must be converted to floats
    V_Mag.append(float(Columns[1]))
    B_V.append(float(Columns[2]))
    Parallaxes.append(float(Columns[3]))
    Parallax_Error.append(float(Columns[4]))
```

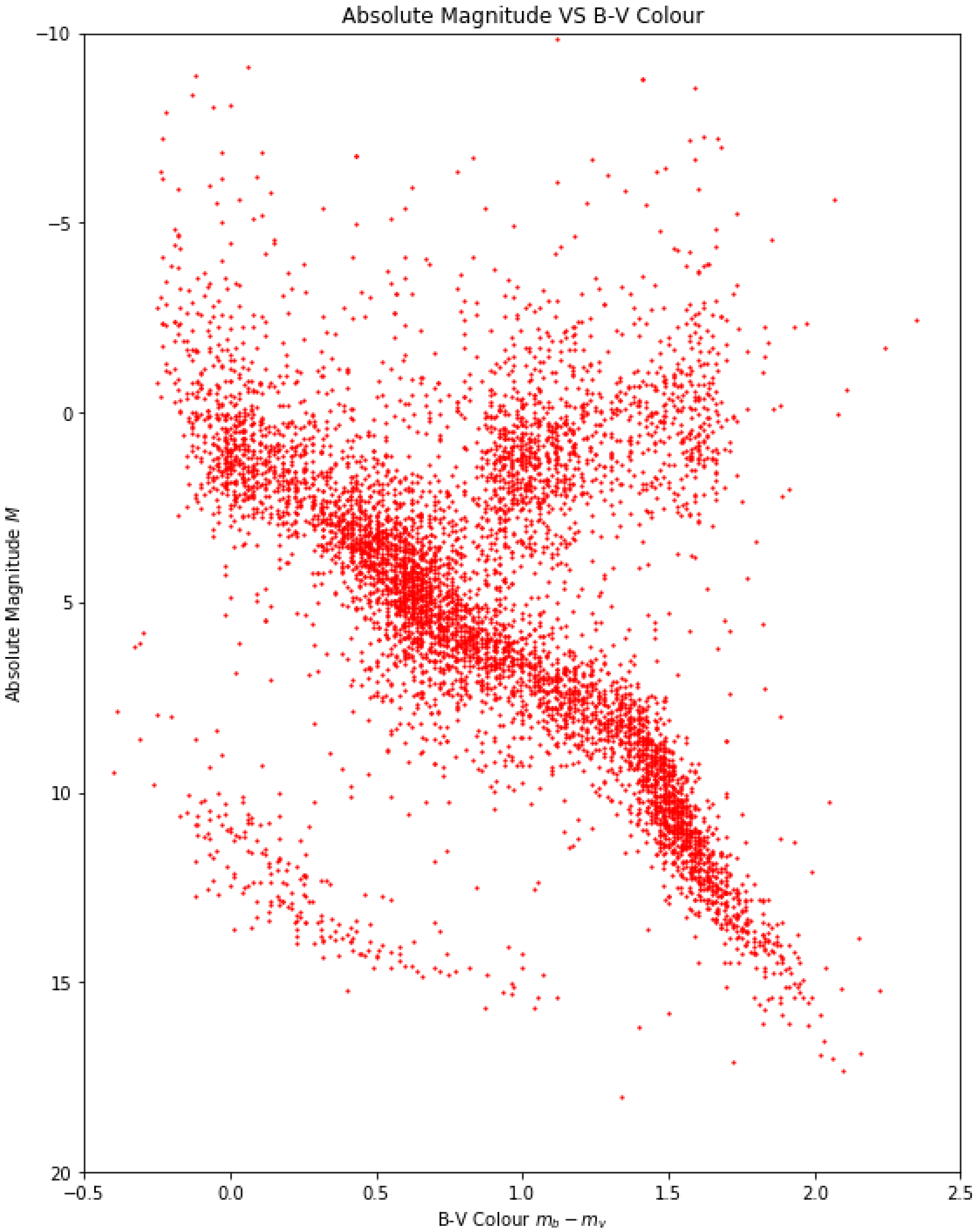
In [27]:

```
#task 2

# NB plt.scatter()
#plot V_Mag on y axis and B_V on the x axis


fig= plt.figure(figsize=(9,12)) #plotting graph
plt.scatter(B_V, V_Mag,s = 1.5 ,c = 'r', marker = 'o')#set point size to 1.5 so all points are visible
plt.title("Apparent Magnitude VS B-V Colour")
plt.xlabel("B-V Colour $m_b - m_v$")
plt.ylabel("Apparent Magnitude $m_v$")
plt.xlim(-0.5, 2.5)
plt.show()
```


Apparent Magnitude VS B-V Colour
Apparent Magnitude m_v
B-V Colour $m_b - m_v$
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

In [37]:

```
#task 3
####  m- M = 5log(d/10)



K = len(V_Mag) #define K as size of new arrays to match the size of the ones imported from the text file

d = np.zeros(K) #d is the distance found from parallax angle p (d = 1/p)
M = np.zeros(K) #M is the absolute magnitude found from:  m- M = 5log(d/10)


for i in range(K):
    d[i] = 1/(float(Parallaxes[i])) #find d for each star from its parallax value
    M[i] = V_Mag[i] - 5*np.log10(d[i]/10) #use formula to find absolute Magnitude for each star

fig= plt.figure(figsize=(9,12)) #plotting graph
plt.scatter(B_V, M,s= 1.5 ,c= 'r', marker = 'o')
plt.title("Absolute Magnitude VS B-V Colour")
plt.xlabel("B-V Colour $m_b - m_v$")
plt.ylabel("Absolute Magnitude $M$")
plt.xlim(-0.5, 2.5)
plt.gca().invert_yaxis() #invert y axis as Hertzsprung Russ
plt.ylim(20,-10)
plt.show()


#varibales to store numbers of Main S
MainSeq = 0
RedGiant = 0
WhiteDwarf = 0

#approximate how many Main Sequence, Red Giant and White Dwarfs are on the diagram

for i in range(K):
    if B_V[i] > 0.8 and M[i] < 3 :
        RedGiant += 1
    elif B_V[i] < 1.25 and M[i] > 10:
        WhiteDwarf += 1
    else:
        MainSeq += 1


print("There are approximately", MainSeq, "Main Sequesnce Stars in this diagram ({0:0.2f}%)".format((MainSeq/K *100)))
print("There are approximately", RedGiant, "Red Giant Stars in this diagram ({0:0.2f}%)".format((RedGiant/K *100)))
print("There are approximately", WhiteDwarf, "White Dwarf Stars in this diagram ({0:0.2f}%)".format((WhiteDwarf/K *100)))
```



```
There are approximately 4895 Main Sequesnce Stars in my diagram (78.70%)
There are approximately 1117 Red Giant Stars in my diagram (17.96%)
There are approximately 208 White Dwarf Stars in my diagram (3.34%)
```

```
In [ ]:
```