Coma Cluster virial mass

April 11, 2022

Ben Rasmussen: V00943006

1 Introduction:

Large scale galactic structures exhibit properties that are inconsistent with both the distribution and amount of matter in them, as well as the luminosity profile of the structure. In this paper, this contradiction will be addressed as well as quantitatively analyzed.

The basic assumption that on extremely large length scales, galaxies may act as point particles in the context of a much larger galactic cluster allows us to infer a mass of the entire system using the virial theorem. Along with this assumption, the following analysis requires that the cluster in question is entirely pressure supported. The Virial theorem tells us that:

$$2\langle KE\rangle + \langle U\rangle = 0 \tag{1}$$

The above can be combined with the *half-mass radius* of the system and the velocity dispersion of member galaxies in order to find the virialized mass enclosed within that radius. It can be shown that:

$$M_{1/2} = \frac{3\sigma^2 r_{1/2}}{G} \tag{2}$$

With σ as the velocity dispersion of the galaxies in the cluster. The radius used above is related to the more colloquial effective radius R_{eff} by the following expression:

$$r_{1/2} = \frac{4R_{eff}}{3} \tag{3}$$

In order to achieve the above results, the system must be assumed to be both relaxed <u>*and*</u> virialized. The first assumption tells us that the system is either in equilibrium with itself or within a relaxation time on the order of the quantities studied in the following analysis. More importantly, the second supposition allows us to apply the virial theorem and reduce the problem to a single, near-isolated system. A virialized system requires that the individual interacting particles are stable within the context of the whole. In a system such as this, the virial theorem above is applicable and any dynamical interactions are assumed to be only gravitationally driven.

It then follows that we must assume limited cosmological effects on the system (although the distance scales are large) which is a by-product of the claim that each galaxy acts independent of each other and as test particles. Later on in the analysis it will also become important that the space-time geometry of the problem be Euclidean so as to mitigate any effects of curvature present within the galactic cluster.

In order to find the velocity dispersion present in equation 2, we can borrow some ideas from cosmology using the following. Since we have:

$$l = a(t)\chi(r) \tag{4}$$

where l is the proper distance between a set of particular observers at a fixed cosmic time, $a(t)$ is the time dependent scale factor of the universe, and χ is the comoving distance between the same two observers, or in other words, the proper distance from before but in units of the scale factor $a(t)$. If we differentiate the above expression we arrive at the desired result:

$$\frac{dl}{dt} = \frac{da(t)}{dt}\chi(r) + a(t)\frac{d\chi(r)}{dr} \tag{5}$$

Since we know that $\dot{l} = \frac{dl}{dt} = \frac{\dot{a}}{a}l$, we can let $H(t) = \frac{\dot{a}}{a}$ where $H(t)$ is the hubble constant, and we arrive at the following:

$$\frac{dl}{dt} = H(t)l + a(t)\frac{d\chi(r)}{dr} \tag{6}$$

In this case, due to our assumptions above, for one galaxy $\frac{dl}{dt} = v_{los}$ is our line-of-sight velocity while $H(t)l=v_{rec} $ is the recessional velocity of the entire cluster. This allows us to determine the peculiar velocities of all galaxies of interest by the following:

$$v_{peculiar} = v_{los} - v_{rec_{cluster}} \tag{7}$$

With the line of sight velocity, we can fit a gaussian curve to the spread of the galaxies present within the cluster to determine the velocity dispersion of the cluster. The 1 σ deviation of the gaussian function corresponds to the velocity dispersion in our line of sight.

The only things missing now for the analysis thus far are the particular line of sight velocities for a single galaxy and the recession velocity for the whole cluster. These both can be found using the same technique. If the redshifted spectra of each galaxy is measured using spectroscopic techniques, it is possible to find the radial velocity by manipulating:

$$z = \frac{\Delta\lambda}{\lambda_{rest}} = \frac{v_{los}}{c} \tag{8}$$

With z being the respective redshift values of each galaxy. This same idea can be taken further if we take a systematic mean value of all redshifts in member galaxies such that:

$$\langle z \rangle = \frac{v_{rec}}{c} \tag{9}$$

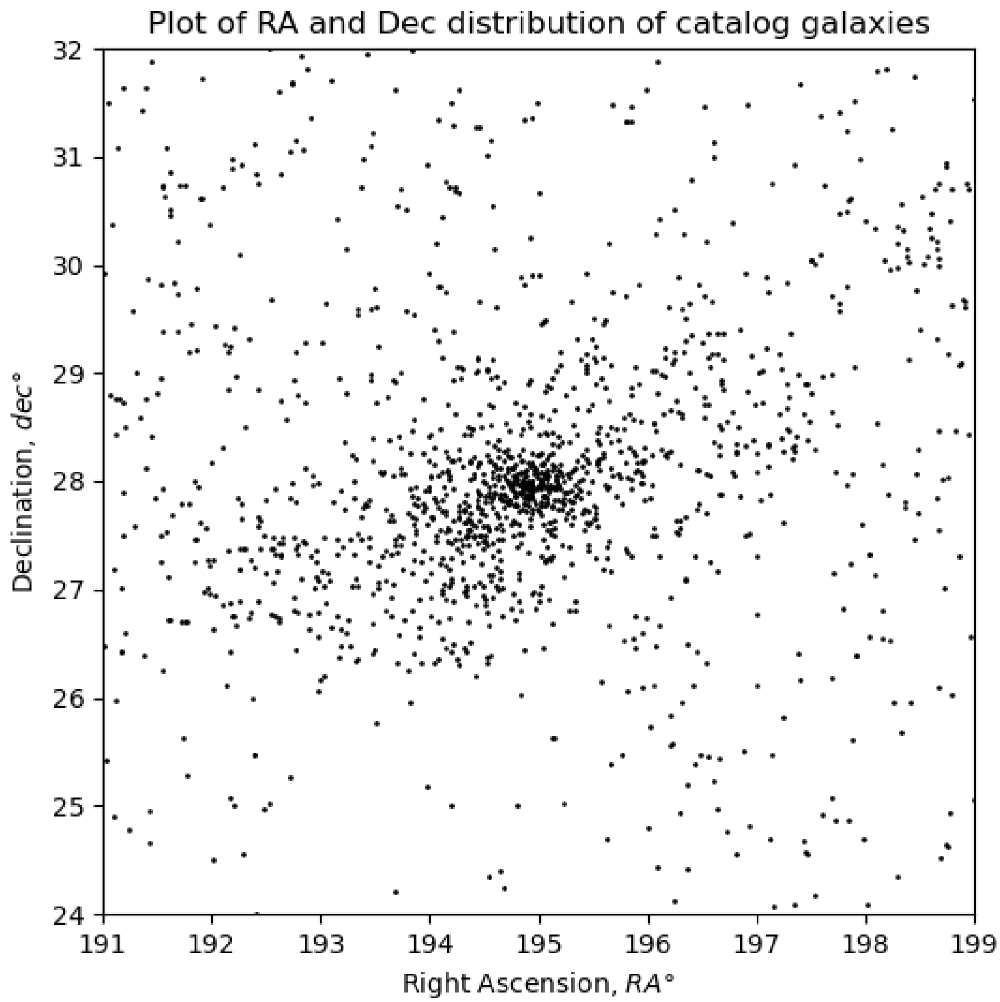
Since we ignored the effects of cosmology up until now, the redshifts used are simply doppler shifts in wavelength rather than cosmological redshifts. Since we are now able to find the *half-mass radius*, $r_{1/2}$, the mass enclosed in that radius using equation 2, $M_{1/2}$ and the velocity dispersion σ it will be possible to find the mass-to-luminosity ratio:

$$\frac{M}{L} = \frac{M_{1/2}}{L_{1/2}} \tag{10}$$

which will necessitate the presence of dark matter in the Coma Cluster as this number will be largely inconsistent with a luminous matter dominated cluster where $\frac{M}{L} \sim 1$.

2 Data/Analysis:

To begin the analysis we must first have data to analyze. The data set was acquired from the *Sloan Digital Sky Survey* data release 16 (SDSS-DR16). Using a bulk search in SQL, 1731 individual galaxies were catalogged with central search positions of $RA = 13:00:00$ and $Dec = 28:00:00$. The spectroscopic class was set to GALAXY while limits on redshift values were set to $0.005 \leq z \leq 0.05$. The initial distribution of galxies was set to a square region of the sky and is visualized as follows:

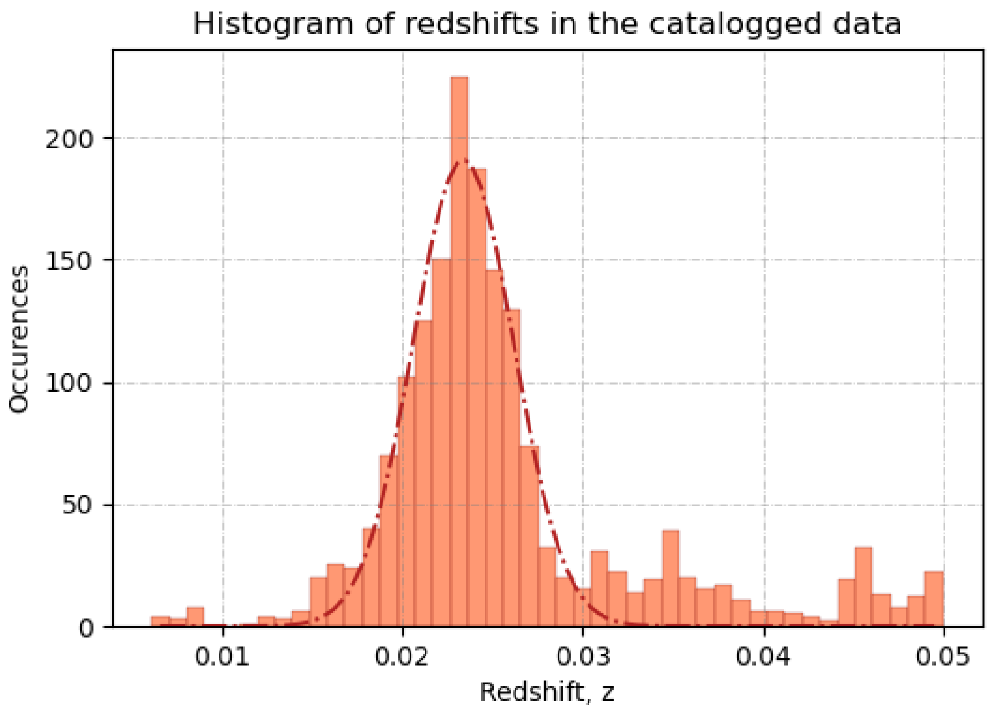


For this lab, all sky positions were either in degrees or radians. It is clear from the above plot that there exists an overdensity of galaxies in a small central region. The sky position of the Coma Cluster was then determined using a weighted mean method such that:

$$\langle \alpha/\delta \rangle = \frac{\Sigma_{i=1}^{n} w_i X_i}{\Sigma_{i=1}^{n} w_i} \tag{11}$$

The weight factors chosen for the galaxy distribution was the relative uncertainty on the r-magnitudes for each galaxy. This was chosen as lower uncertainties on magnitudes likely corresponds to lower uncertainty on relative position in the sky. As such, the weights were applied as the inverse of each particular uncertainty. This was chosen over the apparent magnitude itself as it is more likely that positional uncertainty and magnitudinal uncertainty are more correlated than magnitude with position. A lower magnitude galaxy may not yield a more accurate position measurement. Using this weighting, the initial position values of the cluster (in degrees) was found to be $RA = 194.82$ and $Dec = 28.17$.

A histogram of redshifts was then plotted in order to find an initial mean value. A fitted gaussian function resulted in a central value and uncertainty for the initial measurement and was displayed as follows:



Before filtering the data, the initial redshift value for the coma cluster was found to be: 0.0234 +/- 0.0028. Using this value and equation 9, the recessional velocity of the entire cluster was then found to be 7010.69 km/s. We could then use Hubble's law to convert this recessional velocity to a distance to the cluster:

$$v = H_0 d \longrightarrow d = \frac{v}{H_0} \tag{12}$$

where H_0 is the Hubble's constant from the *Planck* mission.

At this point, the assumption that the universe (around the Coma Cluster) is Euclidean is important as it allows us to use the following formula for angular seperation from a point:

$$\theta \approx \sqrt{[(\alpha - \alpha_0) cos(\delta)]^2 + (\delta - \delta_0)^2} \tag{13}$$

Where α is the right ascension of the star and α_0 is the Coma Cluster central RA. This is true, similarly, with declination and δ. The $cos(\delta)$ accounts for the right ascension modulation due to the spherical geometry.

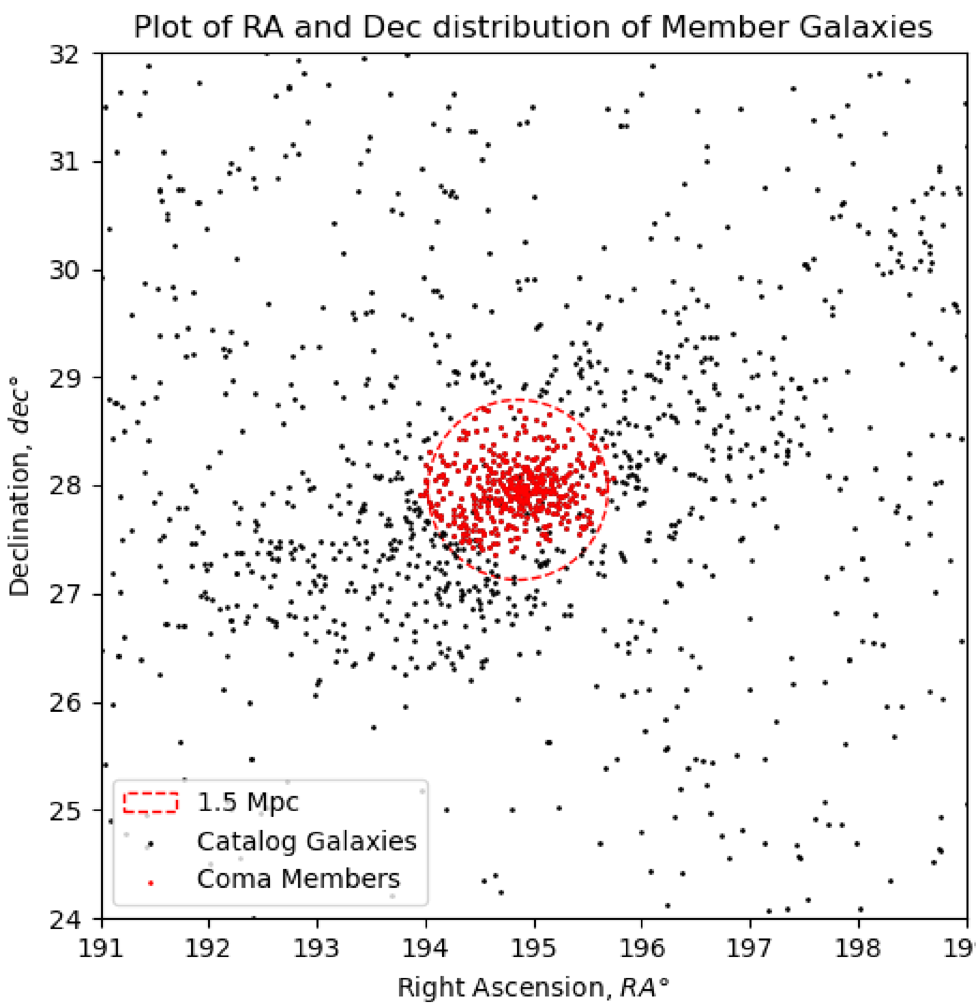
After this, we can then use another small angle approximation in the radial direction with:

$$\theta(rads) = \frac{projected\ radial\ distance\ (R)}{distance\ to\ cluster\ (d)} \tag{14}$$

So we then arrive at a result for the linear seperation in Mpc:

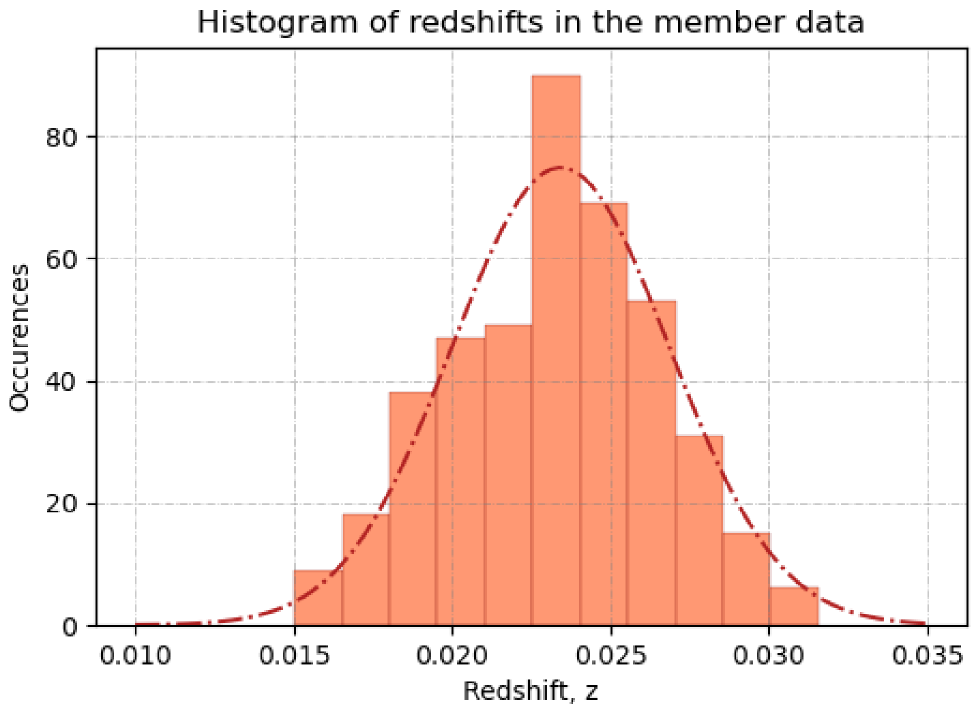
$$R = \theta(rads) \cdot d(Mpc) \tag{15}$$

To improve the data set, a filtering algorithm was applied to the original data. Members of the Coma Cluster are likely within $\sim 1.5Mpc$ from the mean centre of the cluster and so any galaxies beyond that were filtered out. Further, in order to remove both foreground and background galaxies, a filter of $\pm 3\sigma$ was placed on the redshift values for each galaxy from the mean. This was repeated until a convergent number of member galaxies was obtained. In doing the above preceedings, a new subset of the original galaxies were now the primary group of interest and could be visualised in context to the whole data set as:



After convergence was achieved, the final RA and Dec values were found to be $194.86ˆo $ and $27.96ˆo $ respectively. The mean redshift of the galaxies now included in the survey was found to be $0.0234 + / - 0.0034$. The number of galaxies of interest were reduced from 1731 initially, to

425 in the member set. Using this new member set, it was now possible to accurately recreate the histogram from before with better data such that:



Given the r-band magnitudes from the SDSS survey along with the corresponding extinction in that band we then found the absolute r-band magnitudes as well as the error. This was done using the extinction-corrected distance modulus such that:

$$m_{obs} - M = 5log_{10}(r) - 5 + A \tag{16}$$

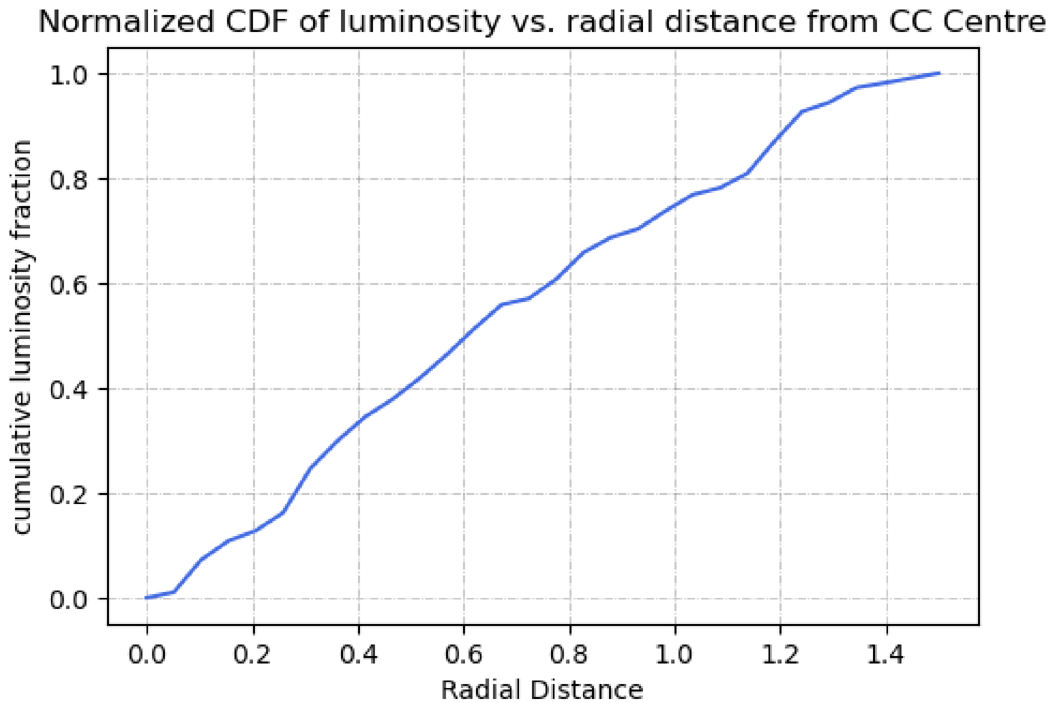
Where A is the extinction term and r is the distance to the coma cluster. This results in an absolute magnitude of each galaxy of:

$$M = -5log_{10}(r) + 5 - A + m_{obs}$$

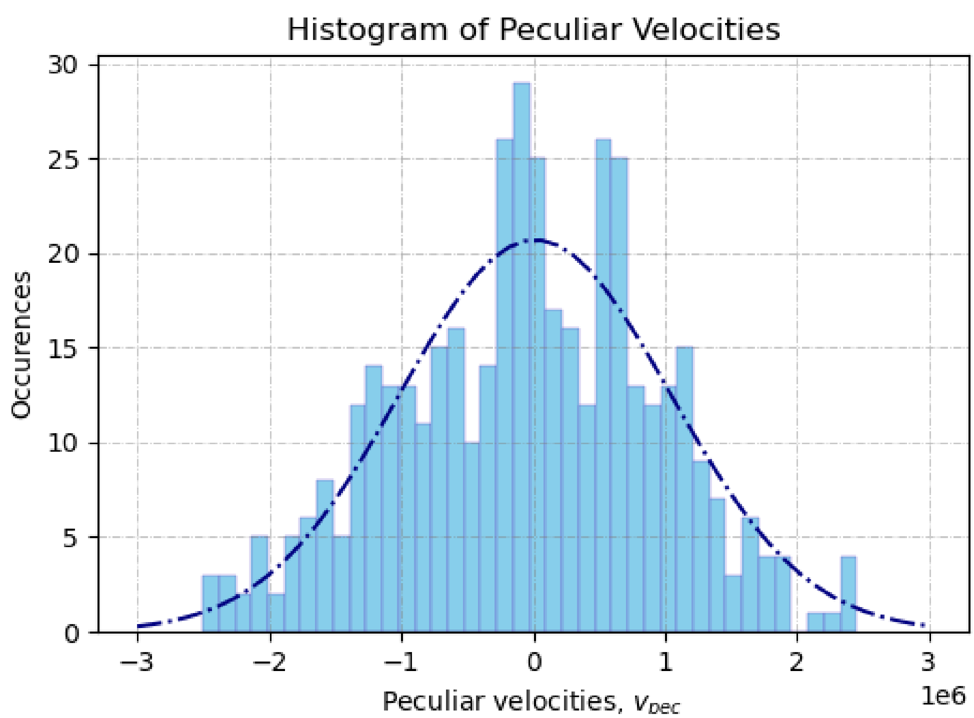
It then follows that the luminosity of each galaxy is given by:

$$L_{gal} = L_{\odot} \cdot 10^{\frac{1}{2.5}(M_{\odot} - M_{gal})} \tag{17}$$

Using these luminosities for the member galaxies we could then plot a normalized cumulative distribution function of luminosity where the normalization factor is the cumulative luminosity at the maximum radius (in this case 1.5 Mpc):



The cumulative luminosity in solar units was found to be $3.682 * 10^{12} L_{\odot}$. Using the above plot and equation 3, a value for both the effective radius and half-mass radius were found to be $R_{eff} = 0.613 Mpc$ and $r_{1/2} = 0.817 Mpc$. The peculiar velocities of all member galaxies were computed using equations 7 and 8 and plotted in a histogram:



With the gaussian fit from above, the line of sight velocity dispersion was finally determined and utilized to find the the mass at the half mass radius using the Virial mass equation, or equation 2. This was found to be $6.00 * 10^{14} M_{\odot}$. By interpolating the normalized CDF, the total luminosity at the *half-mass-radius* was determined. The value for luminosity was $2.3894 * 10^{12} L_{\odot}$. Using equation 10, the ratio between these values was found. The significance of these results will be discussed below but the final mass to luminosity ratio of the Coma Cluster was determined to be $251.24 M_{\odot}/L_{\odot}$ which is far far greater than 1.

3 Discussion:

In the process of obtaining the above results, a number of significant assumptions were made. In order to correctly derive both equations 2 and 3, they rely on the assumption that equation 1 is valid. This implies that the galactic system is both relaxed and virialized, imposing the restrictions that the Coma cluster is barely dynamic and not undergoing significant kinematic change. Although this may not be true on single galactic scales, averaging over the entire cluster seems to confirm that this is relatively valid. If the system was highly dynamic, the velocity dispersion magnitude would be on the order of the recession velocity. As it is ~ 10x smaller, the assumption likely holds. This, however, leads to the initial presumption that galaxies can act as test particles with no extension. We know that the average size of a galaxy falls roughly between $1kpc$ to $100kpc$. Since the filtered size of the Coma Cluster is $1.5Mpc$, we can clearly see that $\langle R_{gal} \rangle << R_{Coma}$. It is then fitting that we can assume point mass bodies for almost all galaxies in the region.

In order to produce the angular seperation results, two key assumptions were made. The first, made rather implicitly, was that the result in equation 13 was dependent on the size of the difference from $(\alpha - \alpha_0)$ and $(\delta - \delta_0)$. We needed to require this value to be $<< 1$ in radians for the equation to be valid. In the appendix the galaxy positions in radians are very similar in magnitude to the central positions and thus the above assumption holds. We also assumed that the spatial geometry of the region was Euclidean within the Cluster. Without further general relativity considerations, this assumptions is largely untested within this paper. As we demonstrated above, the length scales of the problem are of orders higher than a single galactic nucleus and thus a flat system is reasonable.

We also assumed that the shifting effect on the spectroscopic data was entirely due to doppler shifting rather than cosmological redshifting. As the redshift values utilized in this paper are < 0.05, the effects of cosmological expansion are likely overshadowed by the individual and recessional velocities of the galaxies in the Coma Cluster.

The two largest challenges faced in the undertaking of this analysis were determining the weight factors for the mean values of relative position and finding convergence to the final member set of galaxies. The first issue was that the weighting function was ambiguous and left to interpretation. The choice to go with the inverse of uncertainty was largely motivated by a larger perceived relation between magnitude uncertainty and position uncertainty. Although smaller magnitudes would more than likely yield better data, a smaller uncertainty on the same dataset is more indicative of accuracy for the same measure and was chosen as such. In finding convergence towards a final selected dataset, the largest challenge was to find a reasonably fast way to recur the same code. After that was figured out, convergence was determined by a lack of change within the number of total data points. This occured after 6-7 runs of the filtering algorithm. The filtering algorithm successfully reduced the initial 1731 galaxies in the catalog to 425 galaxies within the member group.

If one assumes that both the mass and light output of a sun-like star is relatively normal for most stars, it then follows logically that the ratio of mass in a galaxy to the luminosity in the same galaxy made up of only sun-like stars should necessarily be $\sim 1\ M_\odot/L_\odot$. If we extrapolate this idea to a galactic cluster like the Coma Cluster, even within an order of magnitude to account for an excess of brown dwarfs, we still encounter an incongruency with this basic assumption and the final result. The mass-to-light ratio found in this paper was 251.24 $M_\odot/L_\odot$ which is clearly far far greater than what one would expect. Simple explanations like brown dwarfs or others are simply insufficient to explain the large discrepancy. It then follows that there is a source of matter that neither emits nor interacts with light present in these clusters such that a velocity dispersion like the one found in this paper is possible within our understandings of gravity and the Virial theorem. Since this matter is transparent to the electromagnetic field it is apt to call is dark matter. Therefore, the large discrepancy between the expected and experimental mass-to-light ratios of the Coma Cluster necessitates the presence of dark matter.

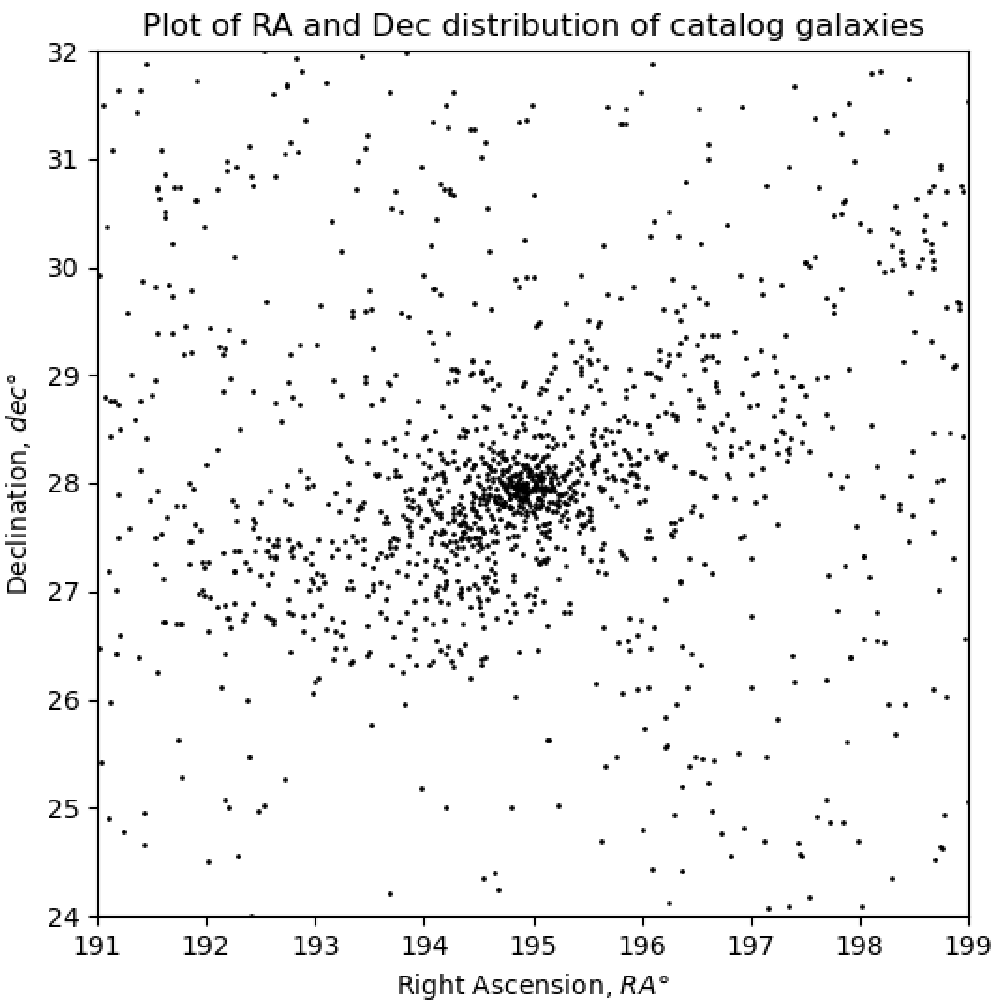
4 Appendix/Code:

[359]:
```
%pylab ipympl
from scipy.optimize import curve_fit
from astropy import constants as const
from astropy import units as u
from matplotlib import patches
from scipy import interpolate

#useful constants:

H0 = 67.8  * u.km / u.s / u.Mpc#km/s/Mpc
Mbol = 4.75 #mags
Mr = 4.76 #mags
Mass = 1.989*10**30 # kg    solar mass
```

```
Populating the interactive namespace from numpy and matplotlib
```

[361]:
```
#loads in the data transposed to rows
#in the following order:
#objID,  ra,  dec,  r,  err_r,  EXTINCTION_r,  z,  zErr

#1:
data = loadtxt('comaclusterdata.csv', comments='#',delimiter=',', skiprows=2).T
datatranspose = data.T

#2:
#plots the RA and Dec sky positions of the included galaxies:

ifig=1; close(ifig);figure(ifig)
scatter(data[1],data[2], c='k',s=0.9)
title('Plot of RA and Dec distribution of catalog galaxies')
xlabel(r'Right Ascension, $RA \degree$')
```

```
ylabel(r'Declination, $dec \degree$')
xlim(191,199)
ylim(24,32)
figsize(6,6)

savefig('DISTRIBUTION.png')
```



Basic weighted mean formula is given by the following:

$$M_W = \frac{\Sigma_{i=1}^{n} w_i X_i}{\Sigma_{i=1}^{n} w_i}$$

In this case, the weighting factor that will be used is the corresponding uncertainty on the r-magnitudes for each galaxy. This is assuming that the resolution on the magnitudes for a particular galaxy is related to the resolution for the position of the same galaxy. As such, the lower uncertainty should yield a lower uncertainty in the position and should be weighted higher. This will be done by taking the inverse of the uncertainty so as to weight smaller values higher.

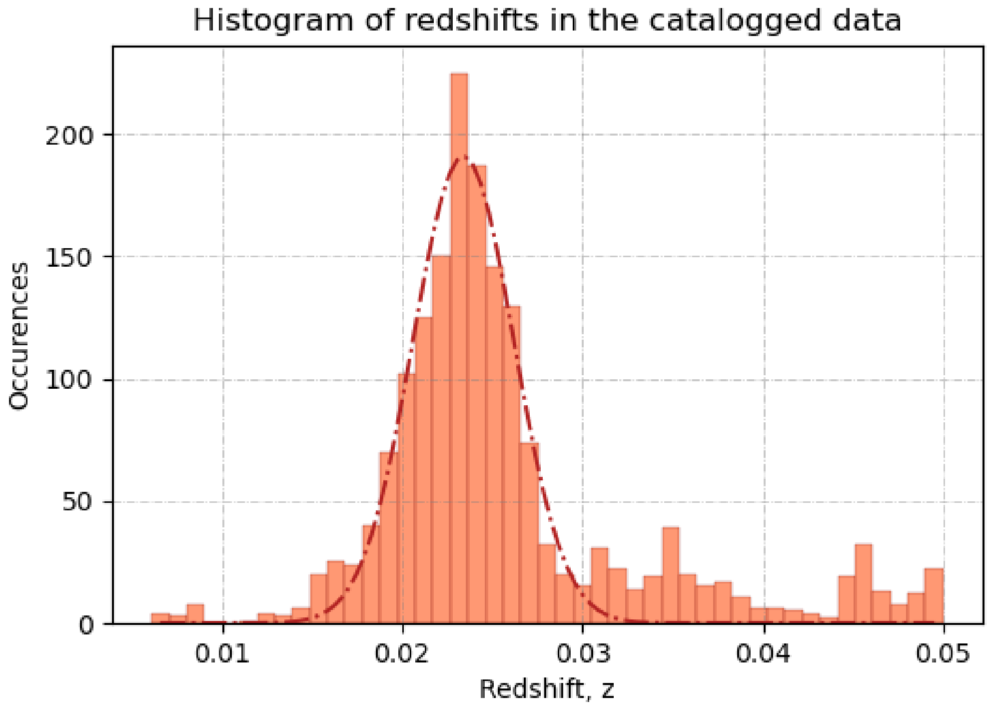
[362]:
```
#3:
#naive means to check weights afterwards
allmeanRA = mean(data[1])
allmeanDec = mean(data[2])

#list of weights and their sum
weights = [data[4][i] **(-1) for i in range(len(data[4]))]
sumweights = sum(weights)

#weighted means for RA and dec
w8edmeanRA = sum([data[1][i]*weights[i] for i in range(len(data[1]))]) /␣
 ↪sumweights
w8edmeanDec = sum([data[2][i]*weights[i] for i in range(len(data[2]))]) /␣
 ↪sumweights

#conversion factor:
d_t_r = 0.0174533 * u.rad
w8edradsRA = w8edmeanRA * d_t_r
w8edradsDec = w8edmeanDec * d_t_r

print(f'The weighted mean values for RA and Dec for the catalog of data are:␣
 ↪{w8edmeanRA:.4f} and {w8edmeanDec:.4f}')
print(f'The weighted mean values for RA and Dec in radians are: {w8edradsRA:.
 ↪4f} and {w8edradsDec:.4f}')
```

```
The weighted mean values for RA and Dec for the catalog of data are: 194.8194
and 28.1666
The weighted mean values for RA and Dec in radians are: 3.4002 rad and 0.4916
rad
```

[364]:
```
#4:

#bin size
Dz = 0.001
#range
zmin,zmax = 0.005,0.05
#number of bins
nbins = int((zmax-zmin)/Dz)
#histogram:
RShist = hist(data[6],nbins ,color='coral')
```

```
#fitting function:
def gauss(x, *p):
    """Gaussian curve to use for fitting the hydrodynamic data,
    takes x values and parameters *p as arguments and returns a
    Gaussian fit
    """
    A, mu, sigma = p
    return A * np.exp(-((x - mu) ** 2) / (2.0 * sigma ** 2))

#initial guess for curve fits:
p0 = [200, 0.025, 0.005]

#points to fit to:
yh=RShist[0]
xh=RShist[1][0:-1]+0.5*diff(RShist[1])



#Gaussian curve fits:
coeff, var_matrix = curve_fit(gauss, xh,yh, p0=p0)

print(f'The initial redshift value for the coma cluster is: {coeff[1]:.4f} +/-␣
↪{coeff[2]:.4f}')



#makes and plots the redshift histogram:

ifig=2;close(ifig);figure(ifig)
RShist = hist(data[6],nbins ,color='coral',edgecolor='maroon',linewidth=0.25,␣
↪alpha=0.8)
xlabel('Redshift, z')
ylabel('Occurences')
title('Histogram of redshifts in the catalogged data')
x = linspace(xh[0],xh[-1],100)
plot(x,gauss(x,*coeff), '-.', c='firebrick')
figsize(6,4)
grid(color ='grey',
        linestyle ='-.', linewidth = 0.5,
        alpha = 0.6)

show()


savefig('redshifthist.png')
```

```
The initial redshift value for the coma cluster is: 0.0234 +/- 0.0028
```



The magnitude of an objects redshift, z, is related to the recessional velocity of the object by:

$$z = \frac{\Delta\lambda}{\lambda_{rest}} = \frac{v_{rec}}{c}$$

where c is the speed of light. The recessional velocity is then related to the distance to the object by Hubble's law:

$$v = H_0 d \longrightarrow d = \frac{v}{H_0}$$

Using the value for H_0 from the *Planck* and the redshift from part 5 we can then find the distance to the Coma Cluster.

[365]:
```python
#5:

#solves for recessional velocity for giving redshift z
def rec_vel(z):
    '''
    Takes redshift z and returns the corresponding recessional velocity
    '''
    v_rec = const.c * z
```

```
    return v_rec

#Solves for the distance to an object given recessional velocity:
def distance(v):
    '''
    Takes recessional velocity and returns the distance to the object
    '''
    d = v / H0
    return d

v_rec1 = rec_vel(coeff[1])
v_reckm = v_rec1.to(u.km / u.s)
d_cc1 = distance(v_reckm)



print(f'The initial found recessional velocity of the Coma Cluster is: {v_reckm:
 ↪.3f}')
print(f'The initial found distance to the Coma Cluster is: {d_cc1:.3f}')
```

```
The initial found recessional velocity of the Coma Cluster is: 7010.690 km / s
The initial found distance to the Coma Cluster is: 103.403 Mpc
```

In order to find the projected radial distances from the center of Coma we first need the RA and dec to be in radians. After that we can use the following small angle formula since we are assuming Euclidean geometry (with negligible cosmological effects):

$$\theta \approx \sqrt{[(\alpha - \alpha_0)cos(\delta)]^2 + (\delta - \delta_0)^2}$$

Where α is the right ascension of the star and α_0 is the Coma Cluster central RA. This is true, similarly, with declination and δ. The $cos(\delta)$ accounts for the right ascension modulation due to the spherical geometry.

After this, we can the use another small angle approximation in the radial direction with:

$$\theta(rads) = \frac{projected\ radial\ distance\ (R)}{distance\ to\ cluster\ (d)}$$

So we then arrive at the desired result for projected radial distance:

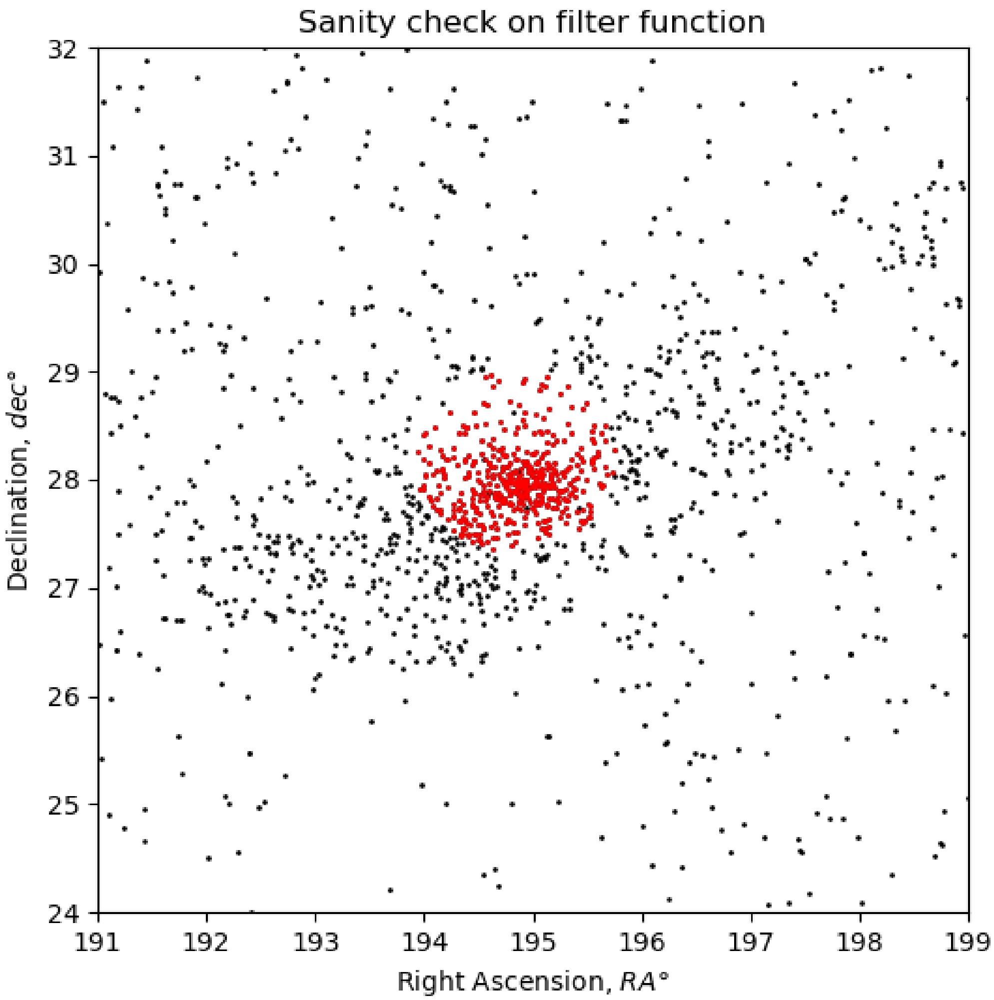
$$R = \theta(rads) \cdot d$$

[366]:
```
#6:

#first we need RA and dec in radians:

#conversion factor:
```

```
d_t_r = 0.0174533 * u.rad

#RA and Dec in radians:
radsRA = data[1] * d_t_r
radsDec = data[2] * d_t_r


#coma cluster position:
a0 = w8edradsRA
d0 = w8edradsDec


#function that evaluates the angular seperation for a particular galaxy given␣
↪centre of CC:

def ang_sep(a,d, a0,d0):
    ''' Takes RA and dec for a galaxy (in radians) as well as the center of the␣
↪coma cluster and
    returns the angular seperation between these points
    '''
    theta = sqrt((((a-a0)*cos(d))**2) + (d-d0)**2)

    return theta

#angular seperations:
angularsep = [ang_sep(radsRA[i],radsDec[i],a0,d0) for i in range(len(radsRA))]

#distance to coma:

d = d_cc1

#function that evaluates the linear seperation between a particular galaxy and␣
↪Coma
#using the angular seperation

def lin_sep(theta, d):
    '''
    Takes angular seperation and distance to the object and returns The linear␣
↪seperation
    '''
    L = theta.value * d
    return L

#linear seperations:
linearsep = [lin_sep(angularsep[i], d) for i in range(len(angularsep))]
```

[368]:
```
#7:
#selects all glaxies within a radial distance of 1.5Mpc from the centre and with
#redshifts within +/- 3 sigma:

#function parameters:
sigma = coeff[2]
z0 = coeff[1]
redshifts = data[6]


def filter_func(L ,z, sigma, z0,i):
    '''
    Takes a linear seperation, redshift, standard deviation of CC mean␣
 ↪redshift, mean redshift and index
    and returns the filtered out indices using L > 1.5 Mpc and (z0-3sigma) < z␣
 ↪< (z0+3sigma)
    '''
    zmin = z0 - 3*sigma
    zmax = z0 + 3*sigma

    if (L.value < 1.5) and (zmin<z<zmax):
        return i
    else:
        pass

#runs the filter function to find indices of galaxies included in catalog
filteredind = [filter_func(linearsep[i],redshifts[i], sigma, z0, i) for i in␣
 ↪range(len(linearsep))]
filteredind = [i for i in filteredind if i]


newdata = []
for i in range(len(filteredind)):
    index = filteredind[i]
    newdata.append(datatranspose[index])

data2 = array(newdata).T

#quick and dirty plot to make sure things are in order
ifig=3; close(ifig);figure(ifig)
scatter(data[1],data[2], c='k',s=0.9)
scatter(data2[1],data2[2], c='r',s=0.9)
title('Sanity check on filter function')
xlabel(r'Right Ascension, $RA \degree$')
ylabel(r'Declination, $dec \degree$')
xlim(191,199)
ylim(24,32)
```

```
figsize(6,6)
```



[369]:
```
#8:
#Now need to repeat the above steps 4 more times to reach convergence to the␣
 ↪true members of the cluster:
#starting data:

#will go about it a different way this time less explicitly:

def convergencefunc(data):
    '''takes a data set with shape 8 by n where n is the number of galaxies␣
 ↪remaining
    in the filter process, does all of the above steps to the data and returns
```

```
    and filtered set of data wit hshape 8 by n where n is the new number of␣
↪galaxies
    '''

    datatranspose2=data.T
    d_t_r = 0.0174533 * u.rad
    #Step 3
    weights = [data[4][i] **(-1) for i in range(len(data[4]))]
    sumweights = sum(weights)

    w8edmeanRA = sum([data[1][i]*weights[i] for i in range(len(data[1]))]) /␣
↪sumweights
    w8edmeanDec = sum([data[2][i]*weights[i] for i in range(len(data[2]))]) /␣
↪sumweights


    w8edradsRA = w8edmeanRA * d_t_r
    w8edradsDec = w8edmeanDec * d_t_r


    #step 4:
    Dz = 0.001
    zmin,zmax = 0.005,0.05
    nbins = int((zmax-zmin)/Dz)
    RShist = hist(data[6],nbins ,color='coral')
    p0 = [200, 0.025, 0.005]
    yh=RShist[0]
    xh=RShist[1][0:-1]+0.5*diff(RShist[1])
    coeff, var_matrix = curve_fit(gauss, xh,yh, p0=p0)

    #step 6:
    radsRA = data[1] * d_t_r
    radsDec = data[2] * d_t_r
    a0 = w8edradsRA
    d0 = w8edradsDec

    angularsep = [ang_sep(radsRA[i],radsDec[i],a0,d0) for i in␣
↪range(len(radsRA))]
    d = d_cc1
    linearsep = [lin_sep(angularsep[i], d) for i in range(len(angularsep))]


    #step 7:
    sigma = coeff[2]
    z0 = coeff[1]
    redshifts = data[6]
```

```
    filteredind = [filter_func(linearsep[i],redshifts[i], sigma, z0, i) for i␣
↪in range(len(linearsep))]
    filteredind = [i for i in filteredind if i]


    newdata = []
    for i in range(len(filteredind)):
        index = filteredind[i]
        newdata.append(datatranspose2[index])

    data2 = array(newdata).T


    return data2


dataint = data2

for i in range(10):
    dataint = dataint
    dataint2 = convergencefunc(dataint)
    if len(dataint2[0]) < len(dataint[0]):
        dataint = dataint2
    else:
        break



memberdata = dataint
```

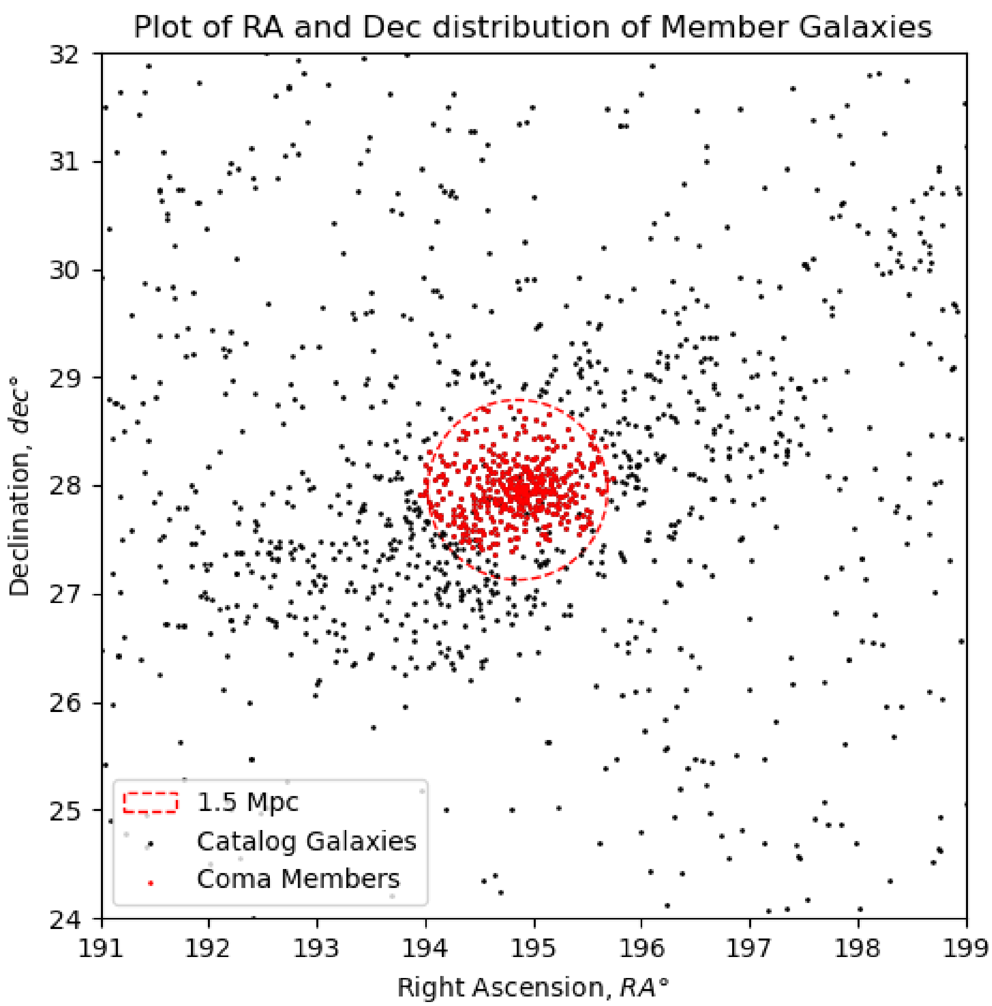
[370]:
```
#Now need RA, Dec and number of member galaxies as well as the mean redshift:

#Number of galaxies:
ngal = len(memberdata[0])
print(f'The number of member galaxies using the above filtering is: {ngal:.0f}')

#RA and Dec positions:

#list of weights and their sum
weights = [memberdata[4][i] **(-1) for i in range(len(memberdata[4]))]
sumweights = sum(weights)

#weighted means for RA and dec
w8edmeanRA = sum([memberdata[1][i]*weights[i] for i in␣
↪range(len(memberdata[1]))]) / sumweights
w8edmeanDec = sum([memberdata[2][i]*weights[i] for i in␣
↪range(len(memberdata[2]))]) / sumweights
```

```
#conversion factor:
d_t_r = 0.0174533 * u.rad
w8edradsRA = w8edmeanRA * d_t_r
w8edradsDec = w8edmeanDec * d_t_r

print(f'The weighted mean values for RA and Dec for the member data is: {w8edmeanRA:.4f} and {w8edmeanDec:.4f}')
print(f'The weighted mean values for RA and Dec in radians are: {w8edradsRA:.4f} and {w8edradsDec:.4f}')


#mean redshift of the cluster:

meanz = mean(memberdata[6])

print(f'The mean redshift of member galaxies in Coma are: {meanz:.4f}')
```

```
The number of member galaxies using the above filtering is: 425
The weighted mean values for RA and Dec for the member data is: 194.8600 and
27.9563
The weighted mean values for RA and Dec in radians are: 3.4009 rad and 0.4879
rad
The mean redshift of member galaxies in Coma are: 0.0232
```

[371]:
```
#9:
#now need to recreate the plot from step 2 with a 1.5Mpc circle surrounding member galaxies

#we need 1.5 Mpc in degrees for the radius of the circle:

radius = 1.5* u.Mpc / d / d_t_r


figure, axes = plt.subplots()
circle = patches.Circle((w8edmeanRA,w8edmeanDec), radius.value, color='r', linestyle = '--',fill=False, label='1.5 Mpc')
axes.set_aspect(1)
axes.add_artist(circle)
scatter(data[1],data[2], c='k',s=0.9, label='Catalog Galaxies')
scatter(memberdata[1], memberdata[2], c='r', s=0.9, label='Coma Members')
title('Plot of RA and Dec distribution of Member Galaxies')
xlabel(r'Right Ascension, $RA \degree$')
ylabel(r'Declination, $dec \degree$')
legend()
xlim(191,199)
ylim(24,32)
```

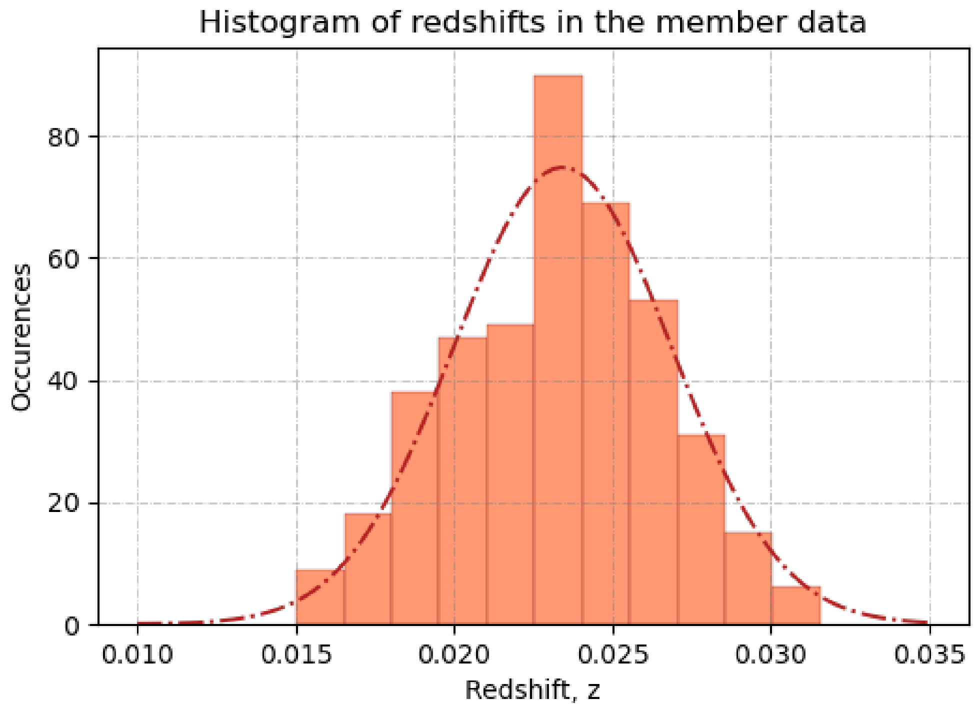
```
figsize(6,6)

savefig('DISTRIBUTIONMEMBERS.png')
```



[373]:
```
%pylab ipympl
#Now need to recreate the histogram plot from step 4:

#bin size
Dz = 0.004
#range
zmin,zmax = 0.005,0.05
#number of bins
nbins = int((zmax-zmin)/Dz)
```

```
#histogram:
#RShist = hist(memberdata[6],nbins ,color='coral')
#initial guess for curve fits:
p0 = [200, 0.025, 0.005]
#points to fit to:
yh=RShist[0]
xh=RShist[1][0:-1]+0.5*diff(RShist[1])



#Gaussian curve fits:
coeff, var_matrix = curve_fit(gauss, xh,yh, p0=p0)

print(f'The final redshift value for the coma cluster is: {coeff[1]:.4f} +/-␣
 ↪{coeff[2]:.4f}')


#makes and plots the redshift histogram:

ifig=5;close(ifig);figure(ifig)
RShist = hist(memberdata[6],nbins ,color='coral',edgecolor='maroon',linewidth=0.
 ↪25, alpha=0.8)
xlabel('Redshift, z')
ylabel('Occurences')
title('Histogram of redshifts in the member data')
x = linspace(0.01,0.035,100)
plot(x,gauss(x,*coeff), '-.', c='firebrick')
figsize(6,4)
grid(color ='grey',
        linestyle ='-.', linewidth = 0.5,
        alpha = 0.6)

show()

savefig('memberredshifthist.png')
```

```
Populating the interactive namespace from numpy and matplotlib
The final redshift value for the coma cluster is: 0.0234 +/- 0.0034
```



Given the r-band magnitudes from the SDSS survey along with the corresponding extinction in that band we can find the absolute r-band magnitudes as well as the error. This is done using the extinction-corrected distance modulus such that:

$$m_{obs} - M = 5log_{10}(r) - 5 + A$$

Where A is the extinction term and r is the distance to the coma cluster. This results in an absolute magnitude of each galaxy of:

$$M = -5log_{10}(r) + 5 - A + m_{obs}$$

[374]:
```
#10:
#Using the SDSS r-magnitudes and extinction values we now need to find the
#Absolute r magnitudes for the galaxies in coma:



def absolute(mobs, A, r=d_cc1):
    '''
    Takes a apparent magnitude, dust extinction in that band and the distance␣
 ↪to
    and object and returns the absolute magnitude of that object
    '''
```

```
    M = -5*log10(r.value *1000000 ) + 5 - A + mobs

    return M

#absolute magnitudes
absmags = absolute(memberdata[3] ,memberdata[5])

#errors on absolute magnitudes
#estimate on uncertainty by finding mean of maximum and minimum values possible␣
 ↪with
#provided uncertainty of m_r

abserrneg = absolute(array(memberdata[3]) + array(memberdata[4]),memberdata[5])␣
 ↪- array(absmags)
abserrpos = absolute(array(memberdata[3]) - array(memberdata[4]),memberdata[5])␣
 ↪- array(absmags)
aberror = (abserrpos - abserrneg)/2
```

We can then turn the absolute magnitudes of each member galaxy into a luminosity in terms of the solar luminosity using the following:

$$L_{gal} = L_{\odot} \cdot 10^{\frac{1}{2.5}(M_{\odot} - M_{gal})}$$

[375]:
```
#11:
#Now need to find luminosities of member galaxies using the above equation
#This will be in terms of the solar luminosity

def luminosity(Mabs, msun = Mr):
    '''Takes the absolute magnitude of a galaxy in the r-band and the solar␣
 ↪magnitude
    in the r-band and returns the luminosity in solar units
    '''
    Lgal = 10**((1/2.5) *(msun - Mabs)) #*L_sun

    return Lgal

#galaxy luminosities:

lumingal = luminosity(absmags) * u.L_sun
print(len(lumingal))
```

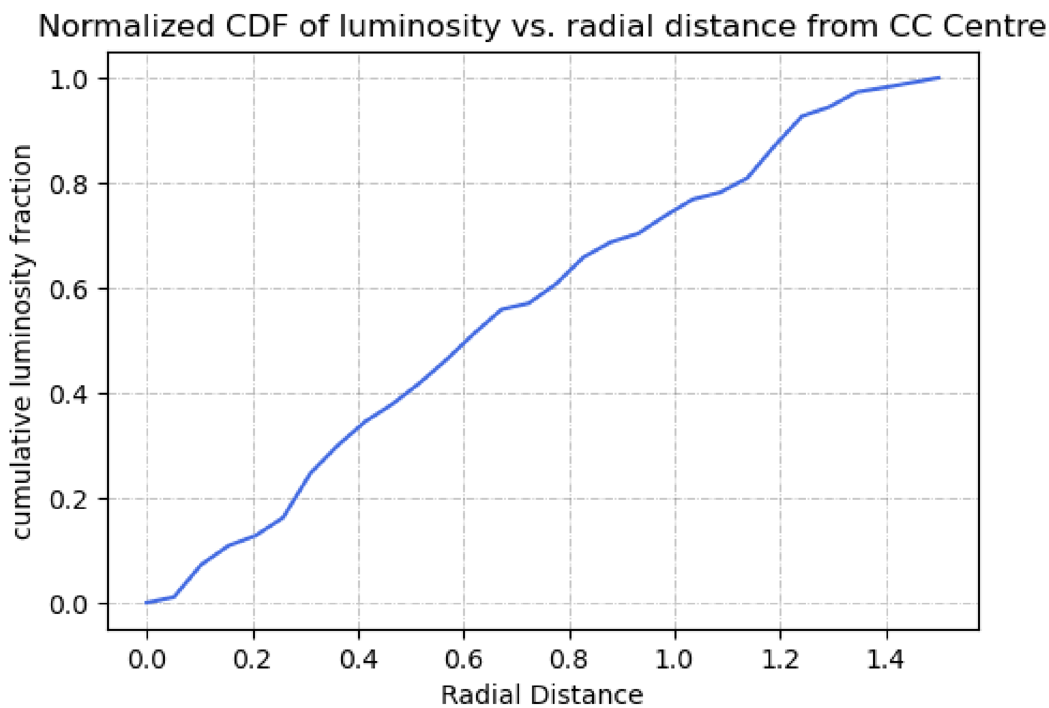
425

[376]:
```
#12:
#now we need both a cumulative luminosity function and the total luminosity
#for the cluster with R<1.5Mpc:
```

```
#Cumulative luminosity is simply just the sum of all luminosity from the member␣
↪galaxies:
cumlumin = sum(lumingal)

print(f'The cumulative luminosity in solar units is: {cumlumin:.5g}')


#we need the radial distance of each star from the centre of the cluster:
radianRA = memberdata[1] * d_t_r
radianDec = memberdata[2] * d_t_r
a0 = w8edradsRA
d0 = w8edradsDec
angles = [ang_sep(radianRA[i],radianDec[i],a0,d0) for i in range(len(radianRA))]
d = d_cc1
R = [lin_sep(angles[i], d) for i in range(len(angles))]

#now need to seperate the radial distance into shells:
shelllumin =[]

#shell size
rs = linspace(0,1.5,30)

#cumulative luminosity as a function of radial shells
for ind1, i in enumerate(rs):
    shellrad = i
    cumlum=0
    for ind2, r in enumerate(R):
        lum = lumingal[ind2]
        radius = r.value
        if radius < shellrad:
            cumlum += lum.value
    shelllumin.append(cumlum)


#normalized luminosity shells
normshelllumin = array(shelllumin)/cumlumin


#plots the figure
ifig=6; close(ifig); figure(ifig)
plot(rs,normshelllumin, c='royalblue',)
title('Normalized CDF of luminosity vs. radial distance from CC Centre')
xlabel('Radial Distance')
ylabel('cumulative luminosity fraction')
grid(color ='grey',
        linestyle ='-.', linewidth = 0.5,
```

```
        alpha = 0.6)
figsize(6,4)
show()




savefig('cumlumfun.png')
```

```
The cumulative luminosity in solar units is: 3.682e+12 solLum
```



To find the half-mass radius of the system we can relate it to the effective radius R_{eff} such that:

$$r_{1/2} = \frac{4R_{eff}}{3}$$

[377]:
```
#13:
#Now need to interpolate the above CDF in order to find where it takes on a␣
↪value of 0.5

#interpolation:
CDFinterp = interpolate.interp1d(normshelllumin,␣
↪rs,kind='cubic',fill_value="extrapolate")
```

```
#effective radius is then:
Reff = CDFinterp(0.5) * u.Mpc

print(f'The effective radius of the cluster is: {Reff:.5f}')


#now for the half-mass radius:

R_12 =  (4/3) * Reff

print(f'The half-mass radius of the cluster is: {R_12:.5f}')
```

```
The effective radius of the cluster is: 0.61287 Mpc
The half-mass radius of the cluster is: 0.81716 Mpc
```

As before we have:

$$z = \frac{\Delta\lambda}{\lambda_{rest}} = \frac{v_{los}}{c}$$

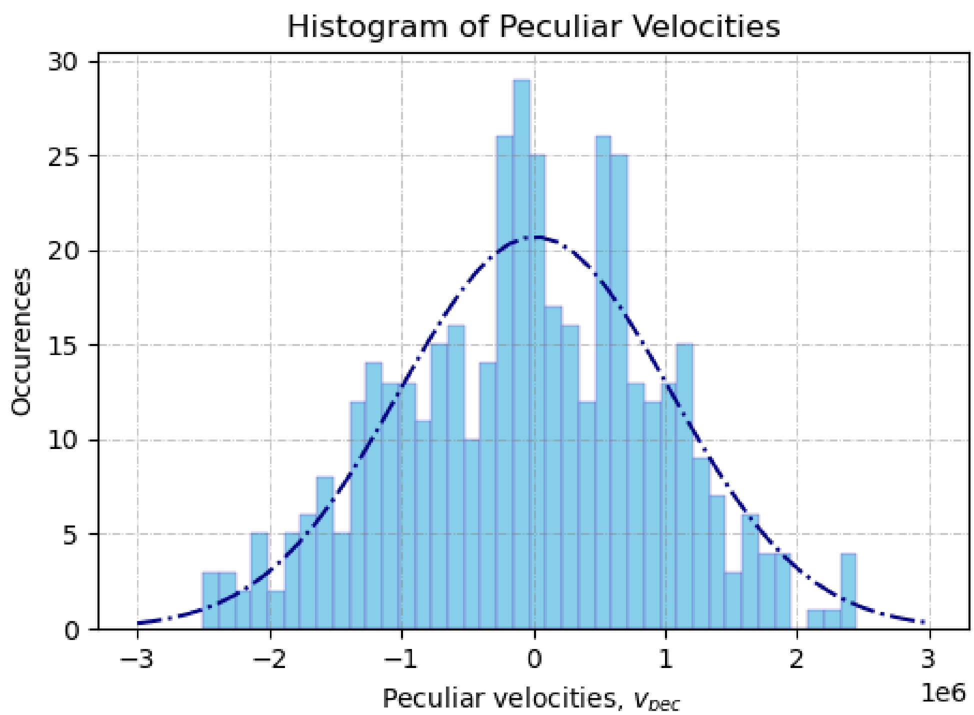
Also, the line of sight velocities are just the sum:

$$v_{los} = v_{rec_{cluster}} + v_{peculiar}$$

This allows for the determination of the peculiar velocities of all of the galaxies.

[378]:
```
#14:
#next, using the redshifts of the member galaxies we need the line of sight
#velocities using the doppler formula from before.


#solves for line of sight velocity for given redshift z:
def los_vels(z):
    '''
    Takes redshift z and returns the corresponding line of sight velocity
    '''
    v_los = const.c * z
    return v_los

#line of sight velocities for member galaxies:
los_vels = los_vels(memberdata[6])



#peculiar velocities now:
pec_vels = (array(los_vels.value) - v_rec1.value) * u.m /u.s
```

[379]:
```
#15:
#Now need to plot a histogram of the above peculiar velocities and subsequently
#fit a gaussian function:




#number of bins
nbins = 40
#histogram:
RShist = hist(pec_vels.value,nbins ,color='coral')
#initial guess for curve fits:
p0 = [25, 0, 25e6]
#points to fit to:
yh=RShist[0]
xh=RShist[1][0:-1]+0.5*diff(RShist[1])

#Gaussian curve fits:
fit, var_matrix = curve_fit(gauss, xh,yh, p0=p0)

#makes and plots the peculiar velocity histogram:
ifig=8;close(ifig);figure(ifig)
RShist = hist(pec_vels.value,nbins ,color='skyblue', edgecolor='slateblue',␣
 ↪linewidth=0.3)
xlabel(r'Peculiar velocities, $v_{pec}$ ')
ylabel('Occurences')
title('Histogram of Peculiar Velocities')
x = linspace(-3e6,3e6)
plot(x,gauss(x,*fit), '-.', c='navy')
figsize(6,4)
grid(color ='grey',
        linestyle ='-.', linewidth = 0.5,
        alpha = 0.6)

show()

savefig('peculiarhist.png')
```



[380]:
```
#the standard deviation of the above gaussian function is the line of sight
#velocity dispersion:

#velocity dispersion in m/s
dispersion = fit[2] * u.m / u.s

print(f'The dispersion of the member galaxies is: {dispersion:.5g}')
```

```
The dispersion of the member galaxies is: 1.0263e+06 m / s
```

Using the velocity dispersion as well as the previously obtained half-mass radius it is now possible to find the virial mass of the coma cluster such that:

$$M_{1/2} = \frac{3\sigma^2 r_{1/2}}{G}$$

[381]:
```
#16:
#Using the velocity dispersion in m/s, the half mass radius in m and the
#units of G we can now find the half-mass of the virialized system:



#quantities
G = const.G
```

```
R_12_m = R_12.to(u.m)
vsigma = dispersion

#M_12 of the coma cluster:

M_12_kg = (3 *(vsigma**2)*R_12_m)/G

#in solar masses
M_12 = M_12_kg.to(u.M_sun)



print(f'The mass of the Coma Cluster in kg: {M_12_kg:.5g}')
print(f'The mass of the Coma Cluster in solar masses: {M_12:.3g}')
```

```
The mass of the Coma Cluster in kg: 1.1937e+45 kg
The mass of the Coma Cluster in solar masses: 6e+14 solMass
```

[382]:

```
#17:
#We now need to interpolate the normalized CDF of the luminosity at the half
#mass radius in order to find the luminosity within that radius:

#can just reverse the interpolation from pervously:

revCDFinterp = interpolate.interp1d(rs,normshelllumin,␣
↪kind='cubic',fill_value="extrapolate")

#at the half mass radius:


L_12_norm = revCDFinterp(R_12)

#unnormalized:

L_12 = L_12_norm * cumlumin

print(f'The half-mass luminosity of the cluster is: {L_12:.5g}')
```

```
The half-mass luminosity of the cluster is: 2.3894e+12 solLum
```

[383]:

```
#18:
#Finally, the ratio of the virial mass and the luminosity of that mass results
#in the mass-to-light ratio of the cluster:


M_L = M_12 / L_12
```

```python
print(f'The mass to luminosity relation of the Coma Cluster is: {M_L:.3f}')

print("This is a very large number compared to ~ 1 as expected. yay!")
```

```
The mass to luminosity relation of the Coma Cluster is: 251.241 solMass / solLum
This is a very large number compared to ~ 1 as expected. yay!
```