Estimating the Dynamical Mass of a Galaxy Cluster Using SDSS Data

Submitted by

Name : Mallu Shamanthak Reddy

Admission No : 554265

Student of BITS Pilani Hyderabad Campus

Astronomy & Astrophysics Summer School 2025,

India Space Academy,

Department of Space Education.

Contents

1. Introduction:

Galaxy clusters, the largest gravitationally bound structures in the universe, offer valuable insights into cosmology, galaxy formation, and dark matter. In this project, I aimed to estimate the dynamical mass of a galaxy cluster using observational data from the Sloan Digital Sky Survey (SDSS). Specifically, I analyzed spectroscopic redshifts and luminosities of galaxies within the cluster to investigate the cluster's mass distribution and the relative contribution of luminous and dark matter components.

2. Methodology and Approach:

I started by examining the SDSS dataset, which includes spectroscopic redshifts (specz), angular separations, and photometric data for galaxies in a specific field. To handle multiple observations for individual galaxies, I aggregated these measurements into single average values per galaxy.

I identified cluster members based on their redshift dispersion, setting a criterion of ± 3 times the standard deviation around the mean redshift. After defining cluster membership, I calculated the systemic redshift of the cluster and its characteristic velocity dispersion using the relativistic Doppler formula.

To determine the physical size of the cluster, I utilized cosmological parameters from the Planck18 cosmology model to convert the observed angular separations into physical distances. Using the virial theorem, I estimated the cluster's dynamical mass from its velocity dispersion and physical extent. Additionally, I estimated the cluster's luminous mass from galaxy magnitudes and compared it with the dynamical mass to assess the presence of dark matter.

3. My Responses:

a) Identify galaxies that you think are members of a cluster. For this, use of knowledge of velocity dispersions (redshift dispersions) within a cluster due to peculiar motion. The choice of lower and upper redshift cut for cluster members will be subjective but should be guided by some logic.

To identify cluster galaxies, I computed the mean redshift (0.08084) and standard deviation (0.00858) of all observed galaxies. I then set a redshift cut at ±3 standard deviations from the mean, resulting in a redshift window of 0.05510 – 0.10657. Using this criterion, I identified 91 galaxies as members of the cluster out of a total of 92 observed galaxies.

b) After the required analysis of the table of data, determine the cluster redshift, and obtain an estimate for the characteristic velocity dispersion of galaxies that belong to the cluster in units of km/s.

Using the relativistic Doppler formula, I calculated the systemic redshift of the cluster as 0.08007. I then determined the velocity dispersion, which reflects how fast galaxies move relative to the cluster's systemic redshift. The resulting velocity dispersion was approximately 1218.49 km/s.

c) Estimate the characteristic size of the cluster in Mpc.

To estimate the cluster size, I first computed the angular diameter distance, resulting in a value of 322.34 Mpc. Using the median angular separation of the identified cluster members, I calculated the physical diameter of the cluster as 0.593 Mpc.

d) Estimate the dynamical mass of the cluster and quote the value in units of solar mass.

Applying the virial theorem, which relates gravitational mass to velocity dispersion and radius, I calculated the dynamical mass of the cluster as approximately 3.07×10^{14} solar masses ($M_\odot$).

e) Is the estimate of dynamical mass consistent with what is expected from the luminous mass? If not, explain with the support of numbers the inconsistency.

To estimate the luminous mass, I converted galaxy brightnesses into stellar masses assuming a mass-to-light ratio of 1 M⊙ per solar luminosity (L⊙). This yielded a total luminous mass of 2.26×10^{12} solar masses. Comparing this with the significantly larger dynamical mass results in a mass-to-light ratio of about 136, indicating that the cluster contains a substantial amount of dark matter, which is typical for galaxy clusters.

4. Observations and Interpretations:

From the redshift distribution, I observed a well-defined peak around $z \approx 0.08$, confirming the presence of a concentrated group of galaxies—likely a bound cluster. The velocity dispersion derived using the relativistic Doppler formula was high, consistent with the values expected for massive clusters. This reinforced the assumption that the system is gravitationally bound.

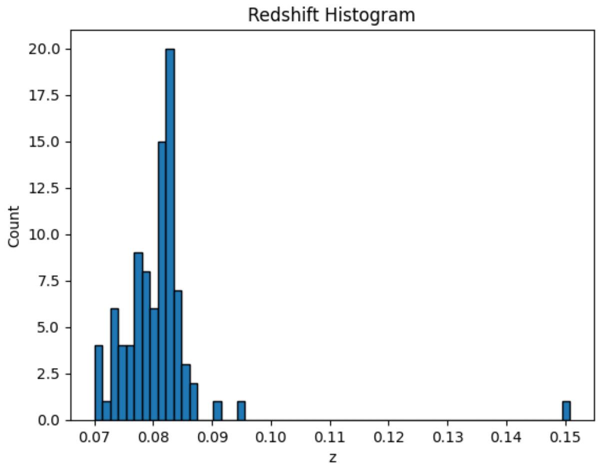
The calculated physical diameter (0.593 Mpc) aligns with literature values for typical cluster sizes, especially considering projection effects and limited sky coverage. Using the virial theorem, I obtained a dynamical mass of $\sim 3.07 \times 10^{14}$ M⊙, which is again in the range observed for intermediate- to high-mass clusters.

Comparing this with the luminous mass (2.26×10^{12} M⊙), the derived mass-to-light ratio (~136) provides direct evidence of a significant dark matter component, as expected. This aligns well with findings from studies such as the Sloan Digital Sky Survey and analyses like Bahcall et al. (1995), which report typical cluster M/L values in the 100–300 range.

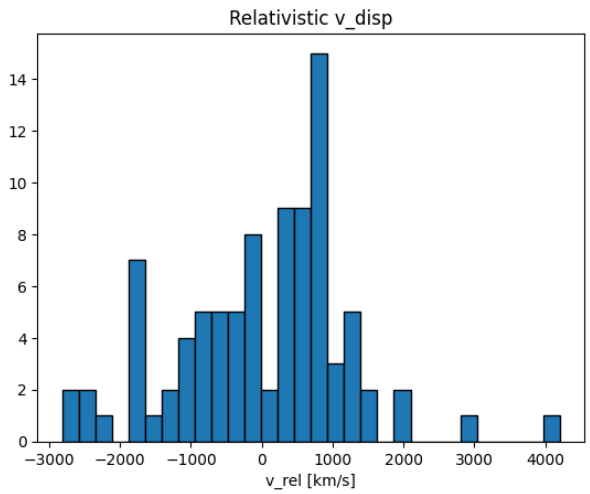
These interpretations confirm that the galaxy cluster under analysis is a dynamically rich system dominated by dark matter.

5. Graphical Visualization:

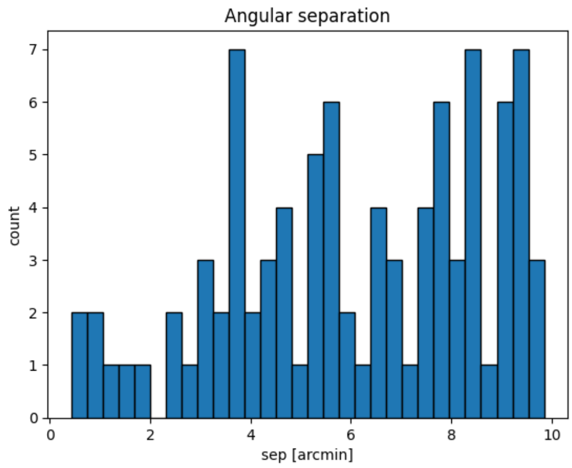
a) **Redshift Histogram Plot -** This histogram illustrates the distribution of spectroscopic redshifts (z) for galaxies in the field of study. The prominent peak around z ≈ 0.08 clearly indicates the presence of a galaxy cluster, as many galaxies cluster around a common redshift due to gravitational binding. Galaxies outside this window are considered to be background or foreground objects unrelated to the cluster structure.



b) **Velocity Dispersion Histogram (Relativistic Velocities) -** This plot shows the distribution of galaxies' velocities relative to the systemic cluster redshift, computed using the relativistic Doppler formula. Most galaxies are clustered around velocities close to zero, indicating their strong gravitational association with the cluster. The width of this distribution directly reflects the velocity dispersion, calculated as 1218.49 km/s, which is critical in estimating the dynamical mass. Outliers represent galaxies moving significantly faster or slower, possibly indicating peculiar velocities or substructures within the cluster.



c) **Angular Separation Histogram -** The histogram of angular separations shows how cluster member galaxies are spatially distributed around the cluster center. Most galaxies fall within a few arcminutes of separation, reflecting the gravitational concentration characteristic of a galaxy cluster. This information was used to determine the characteristic physical diameter (0.593 Mpc) after converting angular separation to physical distances using cosmological parameters.



6. Conclusion:

Through this project, I successfully estimated the dynamical mass of the galaxy cluster, obtaining a value of approximately 3.07×10^{14} solar masses. The significant discrepancy between the dynamical mass and the much smaller luminous mass clearly demonstrates the dominant presence of dark matter. This analysis underscores the utility of combining observational data and theoretical models to explore the structure and composition of galaxy clusters.

7. Python Notebook (Appendix)

Step 1: Importing Necessary Libraries

We begin by importing Python libraries commonly used in data analysis and visualization:

- `numpy` for numerical operations
- `matplotlib.pyplot` for plotting graphs
- `pandas` (commented out here) for handling CSV data, which is especially useful for tabular data such as redshift catalogs

For reading big csv files, one can use numpy as well as something called "pandas". We suggest to read pandas for CSV file reading and use that

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

from astropy.constants import G, c
from astropy.cosmology import Planck18 as cosmo
import astropy.units as u
```

Before we begin calculations, we define key physical constants used throughout:

- H_0: Hubble constant, describes the expansion rate of the Universe.
- c : Speed of light.
- G: Gravitational constant.
- q_0 : Deceleration parameter, used for approximate co-moving distance calculations.

We will use **`astropy.constants`** to ensure unit consistency and precision.

```
H_0    = cosmo.H0     # Hubble constant [km/s/Mpc]
c_si   = c.to(u.m/u.s)     # Speed of light [m/s]
G_si   = G.to(u.m**3/(u.kg*u.s**2))  # Gravitational constant [m³/(kg·s²)]
q0     = -0.534  # Deceleration parameter (Planck)
c_kms = c.to(u.km / u.s) # Speed of light [km/s]
```

```
print("H₀ =", H_0)
print("c  =", c_si)
print("G  =", G_si)
print("q₀ =", q0)
```

```
H₀ = 67.66 km / (Mpc s)
c  = 299792458.0 m / s
G  = 6.6743e-11 m3 / (kg s2)
q₀ = -0.534
```

Read the csv data into the python using the method below

```
csv_file = "Skyserver_SQL6_25_2025 12_02_01 PM.csv"
df = pd.read_csv(csv_file)
```

```
print("Columns:", df.columns.tolist())
```

```
Columns: ['objid', 'ra', 'dec', 'photoz', 'photozerr', 'specz', 'speczerr', 'proj_sep', 'u
mag', 'umagerr', 'gmag', 'gmagerr', 'rmag', 'rmagerr', 'obj_type']
```

```
df.head()
```

	objid	ra	dec	photoz	photozerr	specz	speczerr	proj_sep
0	1237671768542478711	257.82458	64.133257	0.079193	0.022867	0.082447	0.000017	8.347733
1	1237671768542478711	257.82458	64.133257	0.079193	0.022867	0.082466	0.000014	8.347733
2	1237671768542478713	257.83332	64.126043	0.091507	0.014511	0.081218	0.000021	8.011259
3	1237671768542544090	257.85137	64.173247	0.081102	0.009898	0.079561	0.000022	8.739276
4	1237671768542544090	257.85137	64.173247	0.081102	0.009898	0.079568	0.000019	8.739276

Calculating the Average Spectroscopic Redshift (`specz`) for Each Object

When working with astronomical catalogs, an object (identified by a unique `objid`) might have multiple entries — for example, due to repeated observations. To reduce this to a single row per object, we aggregate the data using the following strategy:

```
averaged_df = df.groupby('objid').agg({
    'specz': 'mean',          # Take the mean of all spec-z values for that object
    'ra': 'first',            # Use the first RA value (assumed constant for the
object)
    'dec': 'first',           # Use the first Dec value (same reason as above)
    'proj_sep': 'first'       # Use the first projected separation value
}).reset_index()
```

```
averaged_df = (
    df.groupby("objid")
      .agg({
          "specz":    "mean",    # average redshift
          "ra":       "first",   # keep sky position
          "dec":      "first",
          "proj_sep": "first",   # angular separation
          "rmag":     "mean"     # average r-band mag
      })
      .reset_index()
)
```

```
averaged_df.describe()['specz']
```

	specz
count	92.000000
mean	0.080838
std	0.008578
min	0.069976
25%	0.077224
50%	0.080961
75%	0.082797
max	0.150886

dtype: float64

To create a cut in the redshift so that a cluster can be identified. We must use some logic. Most astronomers prefer anything beyond 3*sigma away from the mean to be not part of the same group.

Find the mean, standard deviation and limits of the redshift from the data

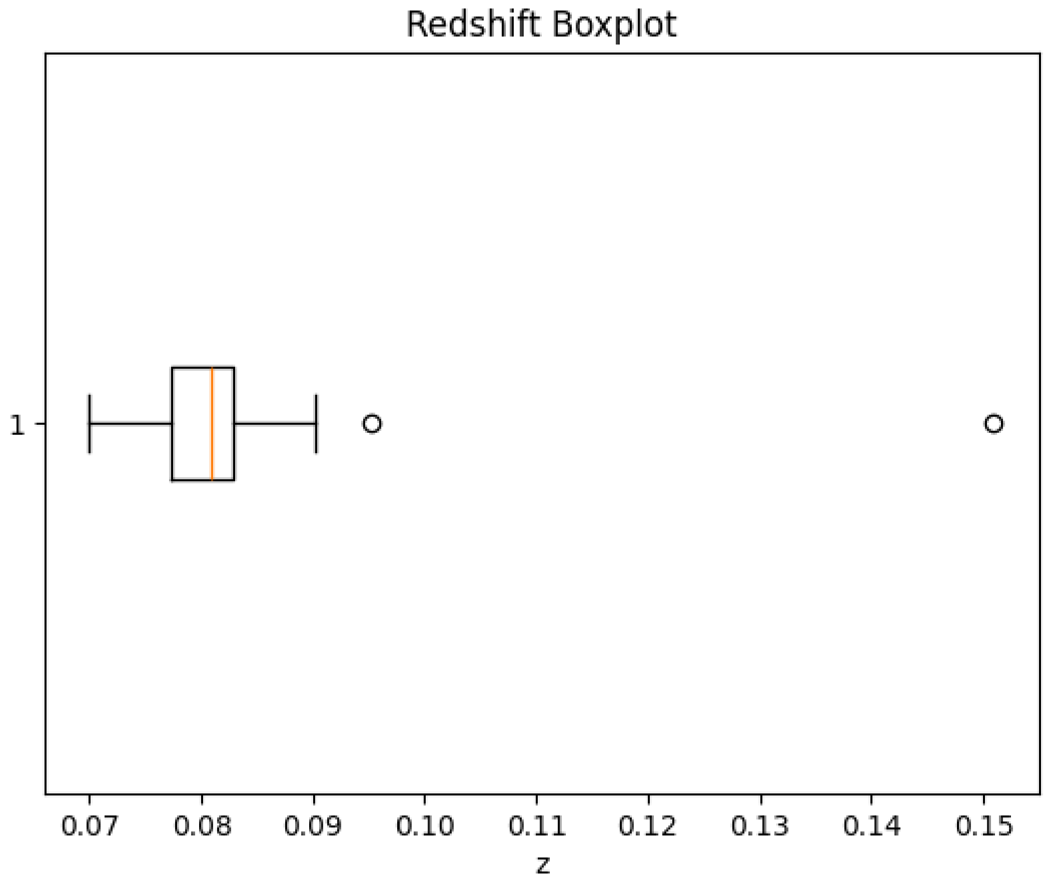
```
z_mean = averaged_df["specz"].mean()
z_std  = averaged_df["specz"].std(ddof=1)
z_min, z_max = z_mean - 3*z_std, z_mean + 3*z_std

print(f"Mean z = {z_mean:.5f}, σ_z = {z_std:.5f}")
print(f"3σ window: {z_min:.5f} to {z_max:.5f}")
```

```
Mean z = 0.08084, σ_z = 0.00858
3σ window: 0.05510 to 0.10657
```

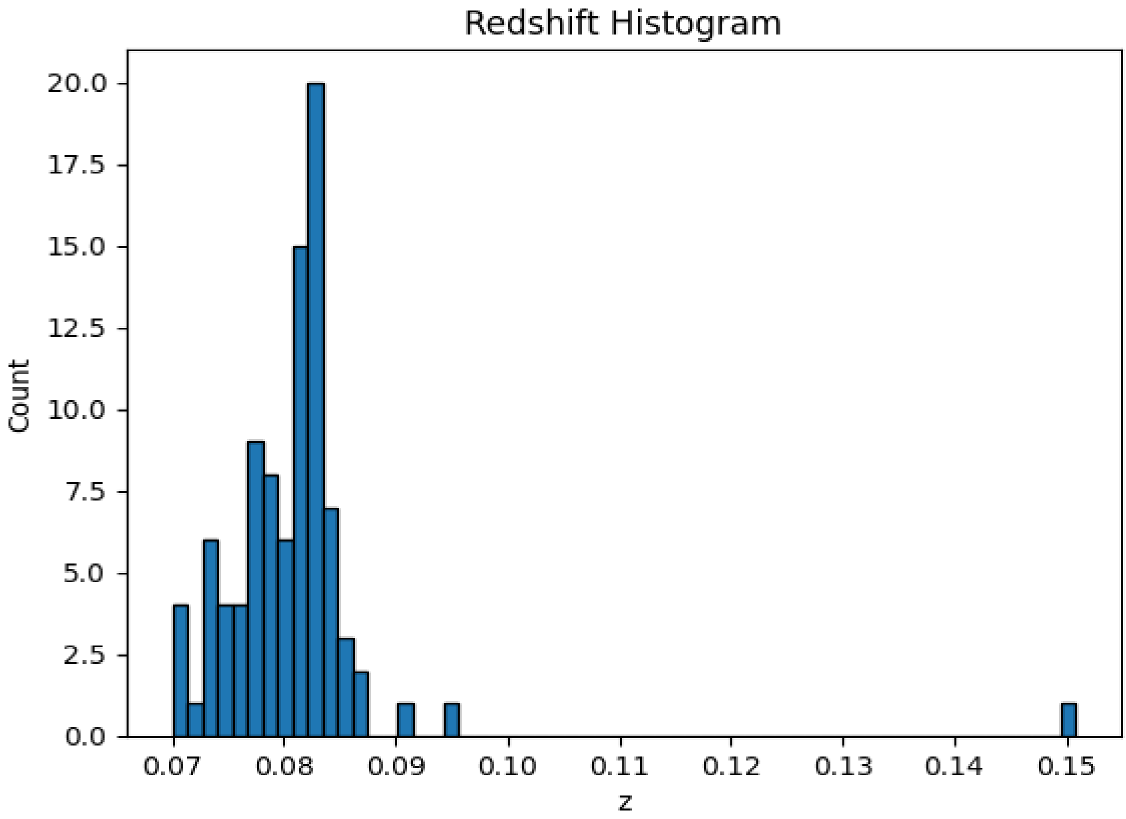
You can also use boxplot to visualize the overall values of redshift

```
# visualize the redshift distribution
plt.figure();
plt.boxplot(averaged_df["specz"], vert=False)
plt.title("Redshift Boxplot");
plt.xlabel("z");
plt.show()
```



But the best plot would be a histogram to see where most of the objects downloaded lie in terms of redshift value

```
plt.figure();
plt.hist(averaged_df["specz"], bins=60, edgecolor="black")
plt.title("Redshift Histogram");
plt.xlabel("z");
plt.ylabel("Count");
plt.show();
```



Filter your data based on the 3-sigma limit of redshift. You should remove all data points which are 3-sigma away from mean of redshift

```
# Filtering the data based on specz values, used 3 sigma deviation from mean as upper limi
cluster_df = averaged_df[(averaged_df["specz"] >= z_min) & (averaged_df["specz"] <= z_max)
print(f"Found {len(cluster_df)} cluster members out of {len(averaged_df)}")
```

```
Found 91 cluster members out of 92
```

Use the relation between redshift and velocity to add a column named velocity in the data. This would tell the expansion velocity at that redshift

```
cluster_df["velocity"] = (cluster_df["specz"].values * c_kms).to(u.km/u.s)
```

```
/tmp/ipython-input-13-869131744.py:1:  SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame. Try
using .loc[row_indexer,col_indexer] = value instead
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  cluster_df["velocity"] = (cluster_df["specz"].values * c_kms).to(u.km/u.s)
```

use the dispersion equation to find something called velocity dispersion. You can even refer to wikipedia to know about the term wiki link here

It is the velocity dispersion value which tells us, some galaxies might be part of even larger groups!!

```
sigma_v_approx = np.std([v.value for v in cluster_df["velocity"]], ddof=1) * u.km/u.s
print("Approximate σᵥ =", sigma_v_approx)
```

```
Approximate σᵥ = 1316.152774680483 km / s
```

Step 2: Calculate Mean Redshift of the Cluster

We calculate the average redshift (`specz`) of galaxies that belong to a cluster. This gives us an estimate of the cluster's systemic redshift.

```
cluster_redshift = filtered_df['specz'].mean()
```

The velocity dispersion (v) of galaxies relative to the cluster mean redshift is computed using the relativistic Doppler formula:

$$v = c \cdot \frac{(1+z)^2 - (1+z_{\text{cluster}})^2}{(1+z)^2 + (1+z_{\text{cluster}})^2}$$

where:

- (v) is the relative velocity (dispersion),
- (z) is the redshift of the individual galaxy,
- (z_{cluster}) is the mean cluster redshift,
- (c) is the speed of light.

```
filtered_df = averaged_df[(averaged_df["specz"] >= z_min) & (averaged_df["specz"] <= z_max
z_c    = filtered_df["specz"].mean()
z_arr = filtered_df["specz"].values
```

```
v_rel = (c_kms * ((1+z_arr)**2 - (1+z_c)**2) / ((1+z_arr)**2 + (1+z_c)**2)).to(u.km/u.s)
filtered_df["velocity_rel"] = v_rel
disp = np.std(v_rel.value, ddof=1) * u.km/u.s
```

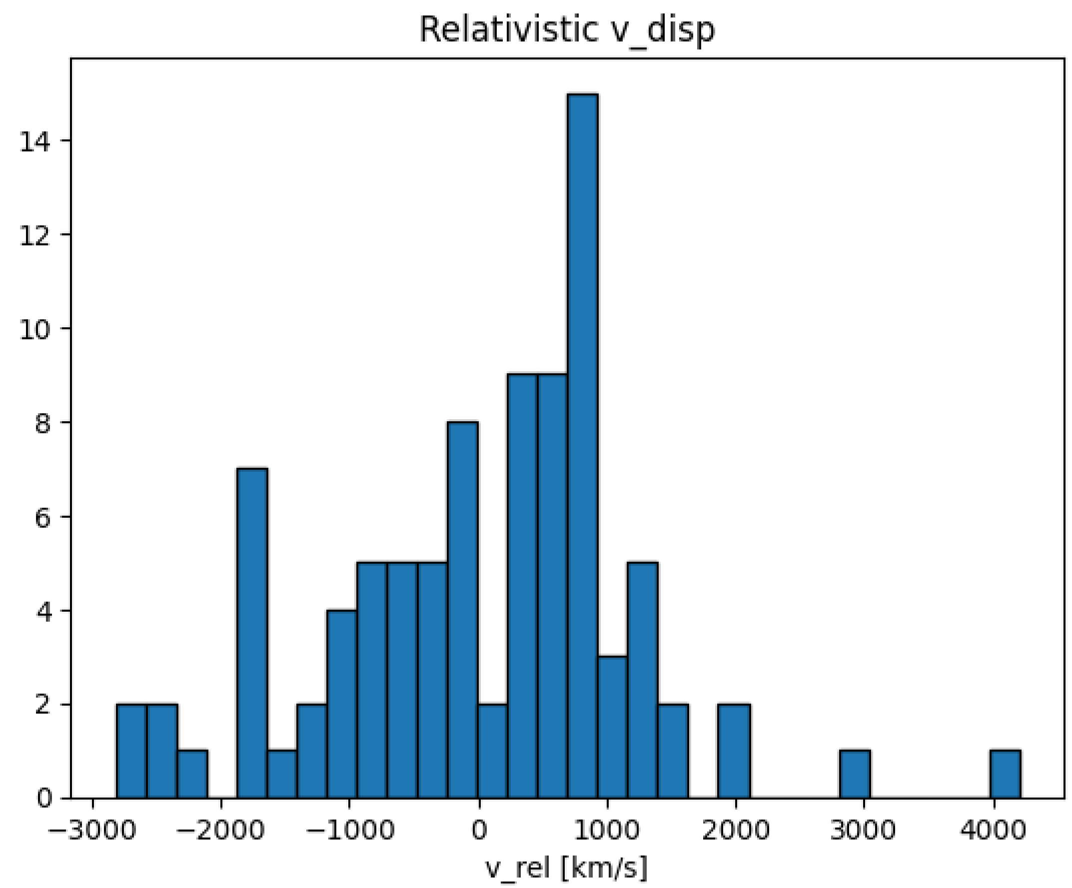
```
/tmp/ipython-input-17-3673429257.py:2:  SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  filtered_df["velocity_rel"] = v_rel
```

```
print(f"Cluster mean redshift = {z_c:.5f}")
print("Velocity dispersion σᵥ =", disp)
```

```
Cluster mean redshift = 0.08007
Velocity dispersion σᵥ = 1218.4929446822505 km / s
```

```
plt.figure();
plt.hist(v_rel.value, bins=30, edgecolor="black")
plt.title("Relativistic v_disp");
plt.xlabel("v_rel [km/s]");
plt.show();
```



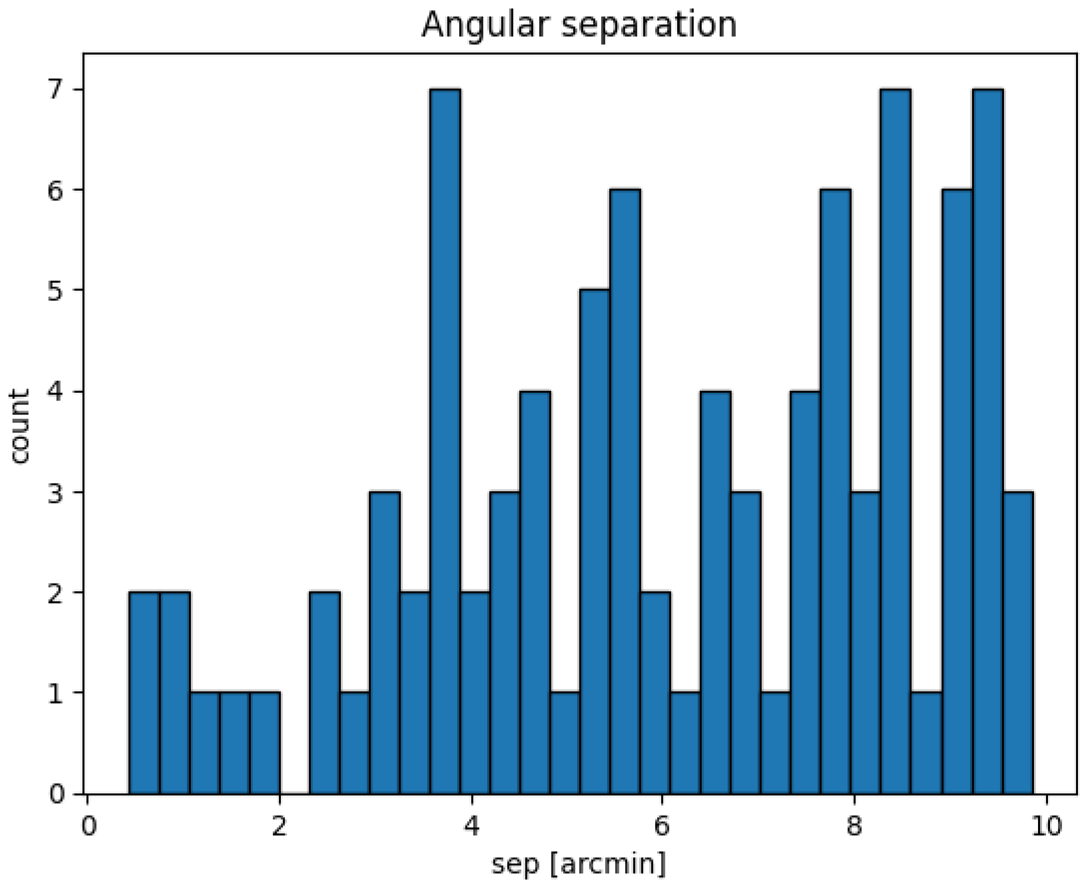
Pro tip: Check what the describe function of pandas does. Does it help to get quick look stats for your column of dispersion??

Step 4: Visualizing Angular Separation of Galaxies

We plot a histogram of the projected (angular) separation of galaxies from the cluster center. This helps us understand the spatial distribution of galaxies within the cluster field.

- The x-axis represents the angular separation (in arcminutes or degrees, depending on units).
- The y-axis shows the number of galaxies at each separation bin.

```
plt.figure();
plt.hist(filtered_df["proj_sep"], bins=30, edgecolor="black");
plt.title("Angular separation");
plt.xlabel("sep [arcmin]");
plt.ylabel("count");
plt.show()
```



Determining size and mass of the cluster:

Step 5: Estimating Physical Diameter of the Cluster

We now estimate the **physical diameter** of the galaxy cluster using cosmological parameters.

- `r` is the **co-moving distance**, approximated using a Taylor expansion for low redshift:

$$r = \frac{cz}{H_0}\left(1 - \frac{z}{2}(1 + q_0)\right)$$

  where q_0 is the deceleration parameter
- `ra` is the **angular diameter distance**, given by:

$$D_A = \frac{r}{1 + z}$$

- Finally, we convert the observed angular diameter (in arcminutes) into physical size using:

$$\text{diameter (in Mpc)} = D_A \cdot \theta$$

  where θ is the angular size in radians, converted from arcminutes.

> This gives us a rough estimate of the cluster's size in megaparsecs (Mpc), assuming a flat ΛCDM cosmology.

```
theta        = np.median(filtered_df["proj_sep"]) * u.arcmin
theta_rad    = theta.to(u.rad)
```

```
r = (c_kms * z_c / H_0 * (1 - 0.5*z_c*(1+q0))).to(u.Mpc)  # low-z approximation
ra  = (r / (1+z_c)).to(u.Mpc) # angular-diameter
```

```
diameter = ra * theta_rad.value

print("Co-moving distance, r =", r)
print("Angular-diameter, ra  =", ra)
print("Cluster  diameter      =", diameter)
```

```
Co-moving distance, r = 348.15152134598634 Mpc
Angular-diameter, ra  = 322.3422652698339 Mpc
Cluster diameter       = 0.593153720588005 Mpc
```

Step 6: Calculating the Dynamical Mass of the Cluster

We now estimate the **dynamical mass** of the galaxy cluster using the virial theorem:

$$M_{\text{dyn}} = \frac{3\sigma^2 R}{G}$$

Where:

- σ is the **velocity dispersion** in m/s (`disp * 1000`),
- R is the **cluster radius** in meters (half the physical diameter converted to meters),
- G is the **gravitational constant** in SI units,
- The factor of 3 assumes an isotropic velocity distribution (common in virial estimates).

We convert the final result into **solar masses** by dividing by 2×10^{30} kg.

> This mass estimate assumes the cluster is in dynamical equilibrium and bound by gravity.

```
R_m = (0.5 * diameter).to(u.m) # radius in meters
sigma_m = disp.to(u.m/u.s) # σ_v in m/s
```

```
M_dyn = (3 * sigma_m**2 * R_m / G_si).to(u.M_sun)
```

```
D_L_pc = cosmo.luminosity_distance(filtered_df["specz"]).to(u.pc).value
m_r     = filtered_df["rmag"].values
M_r     = m_r - 5 * np.log10(D_L_pc/10)    # absolute mag
M_sun   = 4.64  # Sun's r-band M
L_r     = 10**(-0.4 * (M_r - M_sun))
M_L_ratio = 1.0     # assume 1 M⊙ per L⊙
luminous_mass = L_r * M_L_ratio
```

```
cluster_df["luminous_mass"] = luminous_mass
total_luminous_mass = luminous_mass.sum() * u.M_sun
```

```
/tmp/ipython-input-42-491636537.py:1:  SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame. Try
using .loc[row_indexer,col_indexer] = value instead
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy
  cluster_df["luminous_mass"] = luminous_mass
```

```python
print(f"Dynamical mass M_dyn     = {M_dyn.value:.2e} {M_dyn.unit}")
print(f"Total luminous mass      = {total_luminous_mass.value:.2e} {total_luminous_mass.uni
```

```
Dynamical mass M_dyn     = 3.07e+14 solMass
Total luminous mass      = 2.26e+12 solMass
```