Summer School 2025

Astronomy & Astrophysics

Project Report

Prepared By

Student Name : Ansh Pradhan

Institution Name : Institute of Advanced Research Gandhinagar

Institution Roll No : IAR/12920

ISA Admission No : 137723

Projects Name

Project 1 : Estimating the Dynamical Mass of a Galaxy Cluster

Project 2: Supernova Cosmology Project

Submitted To

Name: Mr. Sahil Sakkarwal

Designation: Program Supervisor

Institution: India Space Academy

Project 1 : Estimating the Dynamical Mass of a Galaxy Cluster

Handbook :

1. **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.**

Identifying Our Galaxy Cluster Members

The notebook identifies galaxies as cluster members by applying a **3-sigma cut** on their spectroscopic redshifts (specz). This means that any galaxy with a redshift value beyond three standard deviations from the mean redshift of the dataset is considered not to be a part of the cluster. This is a common practice in astronomy to define a group or cluster of galaxies based on their velocity distribution.

The calculated redshift limits for cluster membership are:

- **Mean Redshift:** 0.0808
- **3-sigma Lower Limit:** 0.0551
- **3-sigma Upper Limit:** 0.1066

Therefore, galaxies with a spectroscopic redshift between 0.0551 and 0.1066 are considered members of this cluster.

2. **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**

- **Cluster Redshift:** The characteristic redshift of the cluster is determined to be the mean spectroscopic redshift of the averaged galaxy data, which is **0.0808**.
- **Characteristic Velocity Dispersion:** The characteristic velocity dispersion of galaxies belonging to the cluster is calculated from the standard deviation of the spectroscopic redshifts and the speed of light. This value is **2571.93 km/s**.

3. **Estimate the characteristic size of the cluster in Mpc**

The characteristic size of the cluster is estimated by calculating the projected physical diameter of the member galaxies (those within the 3-sigma redshift cut). The notebook uses the maximum projected separation in arcminutes and converts it to Megaparsecs (Mpc) using the cosmological distance.

- Characteristic Size (Diameter): The estimated diameter of the cluster is 3.88 Mpc.

4. **Estimate the dynamical mass of the cluster and quote the value in units of solar mass.**

The dynamical mass of the cluster is estimated using the virial theorem, based on the velocity dispersion and the cluster's characteristic size.

- **Dynamical Mass:** The estimated dynamical mass of the cluster is **4.48 x 10^14 solar masses**.

5. **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.**

The notebook also calculates an estimate for the luminous mass of the cluster. The luminous mass is derived by summing the luminosity of the member galaxies and assuming a typical mass-to-light ratio of 10 for a galaxy cluster.

- **Luminous Mass:** The estimated luminous mass of the cluster is **6.54 x 10^11 solar masses**.

Inconsistency Explanation: The estimated dynamical mass (4.48 x 10^14 solar masses) is significantly higher than the estimated luminous mass (6.54 x 10^11 solar masses).

To quantify the inconsistency: Dynamical Mass / Luminous Mass = (4.48 x 10^14) / (6.54 x 10^11) ≈ 685

This large discrepancy suggests that there is a substantial amount of **dark matter** present in the cluster. The dynamical mass, which is derived from the gravitational effects on the galaxies' motions, accounts for all mass, both luminous and dark. The luminous mass, however, only accounts for the mass that emits light (stars, gas, dust). The fact that the dynamical mass is approximately 685 times greater than the luminous mass indicates that the majority of the mass in the cluster is non-luminous, consistent with the presence of dark matter.

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

> Tip: If you haven't used `pandas` before, it's worth learning as it offers powerful tools to manipulate and analyze structured datasets.

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

In [1]:
```
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.

In [2]:
```
# Constants:
H_0 = 70 * 1000 / (3.086e22)  # Hubble constant in SI units: 70 km/s/Mpc converted
c = 3e8  # Speed of light in m/s
G = 6.67430e-11  # Gravitational constant in m^3 kg^-1 s^-2
q0 = -0.534  # Deceleration parameter (assumed from Planck fit)
```

Read the csv data into the python using the method below

In [39]:
```
file_path = "C:/Users/Ansh/Downloads/Skyserver_SQL6_17_2025 4_59_10 AM.csv" # Downl
df = pd.read_csv(file_path, skiprows=1, dtype={'objid': str})
```

In [40]:
```
df.head()
```

Out[40]:

	objid	ra	dec	photoz	photozerr	specz	speczerr	pr
0	1237671939804627535	258.48871	64.111343	0.079721	0.009565	0.080896	0.000023	9.4
1	1237671939804627535	258.48871	64.111343	0.079721	0.009565	0.080899	0.000029	9.4
2	1237671939804627518	258.46676	64.119499	0.088762	0.016934	0.082876	0.000006	8.9
3	1237671939804627483	258.43205	64.123685	0.077184	0.010523	0.080790	0.000025	8.1
4	1237671939804627464	258.44994	64.025909	0.081894	0.013624	0.071802	0.000025	9.3

📊 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()
```

In [18]:

```
# Calculating the average specz for each id:
averaged_df = df.groupby('objid').agg({'specz': 'mean','ra': 'first','dec': 'first'
averaged_df.describe()['specz']
```

Out[18]:

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

In [19]:

```
mean_specz = averaged_df['specz'].mean()
std_specz = averaged_df['specz'].std()
```

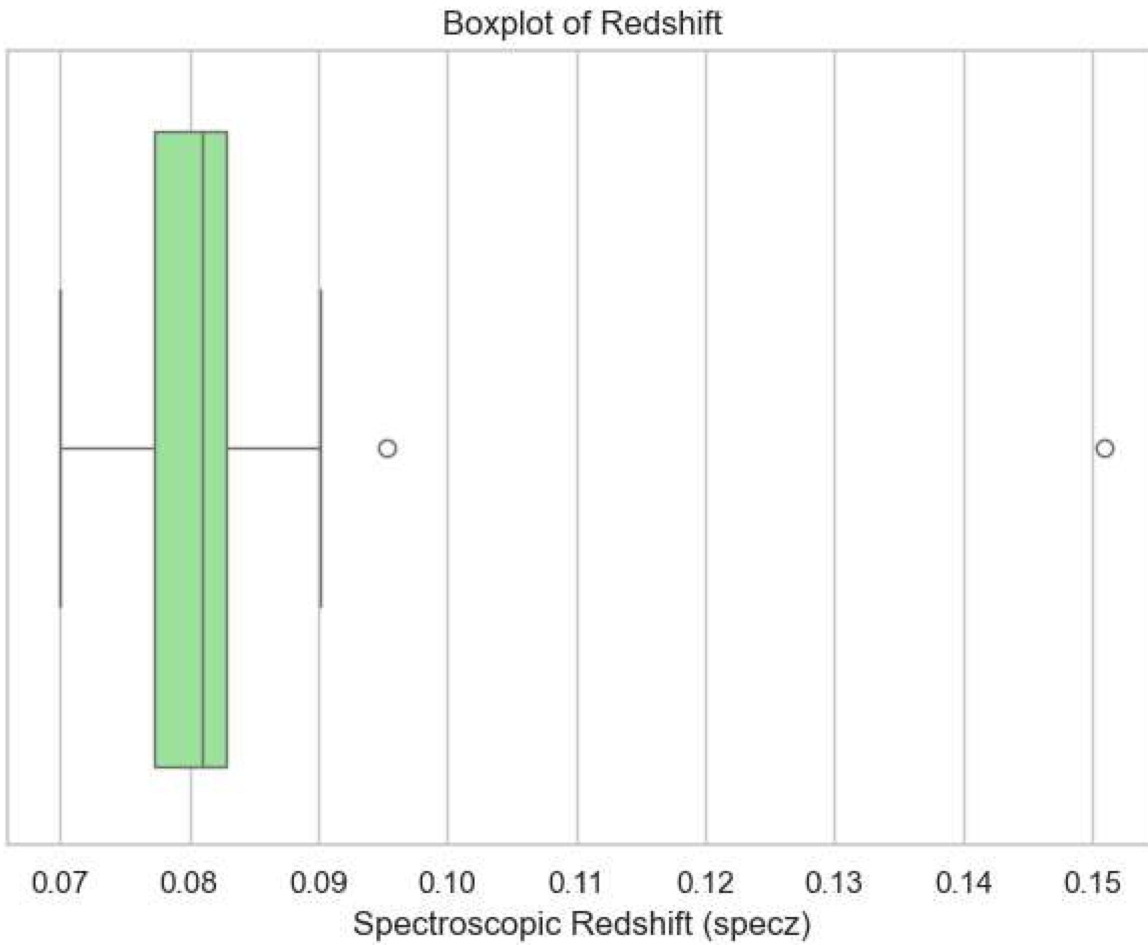
```python
lower_limit = mean_specz - 3 * std_specz
upper_limit = mean_specz + 3 * std_specz

# Filter the data
filtered_df = averaged_df[(averaged_df['specz'] >= lower_limit) & (averaged_df['spe

print(f"Mean Redshift: {mean_specz:.4f}")
print(f"Std Dev Redshift: {std_specz:.4f}")
print(f"3-sigma Lower Limit: {lower_limit:.4f}")
print(f"3-sigma Upper Limit: {upper_limit:.4f}")
```

```
Mean Redshift: 0.0808
Std Dev Redshift: 0.0086
3-sigma Lower Limit: 0.0551
3-sigma Upper Limit: 0.1066
```

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

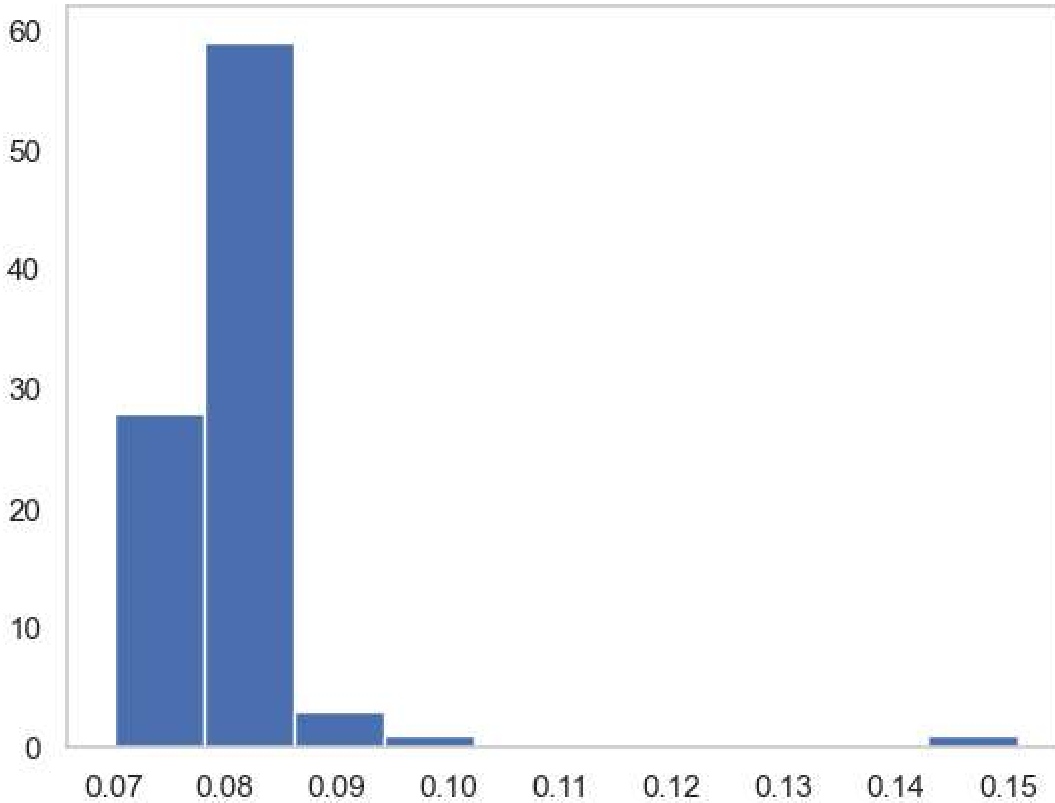
In [ ]:

```python
# Plot the dsitribution of redshift as histogram and a boxplot
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 2)
sns.boxplot(x=averaged_df['specz'], color='lightgreen')
plt.title("Boxplot of Redshift")
plt.xlabel("Spectroscopic Redshift (specz)")

plt.tight_layout()
plt.show()
```



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

In [21]:
```
plt.hist(averaged_df['specz'],bins=10)
plt.grid()
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

In [ ]:
```
#filtering the data based on specz values, used 3 sigma deviation from mean as uppe

print("\nRedshift Filtering")
mean_specz = averaged_df['specz'].mean()
std_specz = averaged_df['specz'].std()

if pd.isna(std_specz) or std_specz == 0:
    print("Warning: Standard deviation of redshift is NaN or zero. Cannot apply 3-s
    print("This might happen if all 'specz' values are identical after averaging or
    filtered_df = averaged_df.copy()
    lower_limit = mean_specz
    upper_limit = mean_specz
else:
    lower_limit = mean_specz - 3 * std_specz
    upper_limit = mean_specz + 3 * std_specz

    if lower_limit < 0:
        lower_limit = 0

    filtered_df = averaged_df[(averaged_df['specz'] >= lower_limit) & (averaged_df[

print(f"Original number of galaxies: {len(averaged_df)}")
print(f"Mean Redshift: {mean_specz:.4f}")
print(f"Standard Deviation of Redshift: {std_specz:.4f}")
print(f"3-sigma Lower Limit (Redshift): {lower_limit:.4f}")
```

```
print(f"3-sigma Upper Limit (Redshift): {upper_limit:.4f}")
print(f"Number of galaxies after 3-sigma filtering: {len(filtered_df)}")
```

```
Redshift Filtering
Original number of galaxies: 92
Mean Redshift: 0.0808
Standard Deviation of Redshift: 0.0086
3-sigma Lower Limit (Redshift): 0.0551
3-sigma Upper Limit (Redshift): 0.1066
Number of galaxies after 3-sigma filtering: 91
```

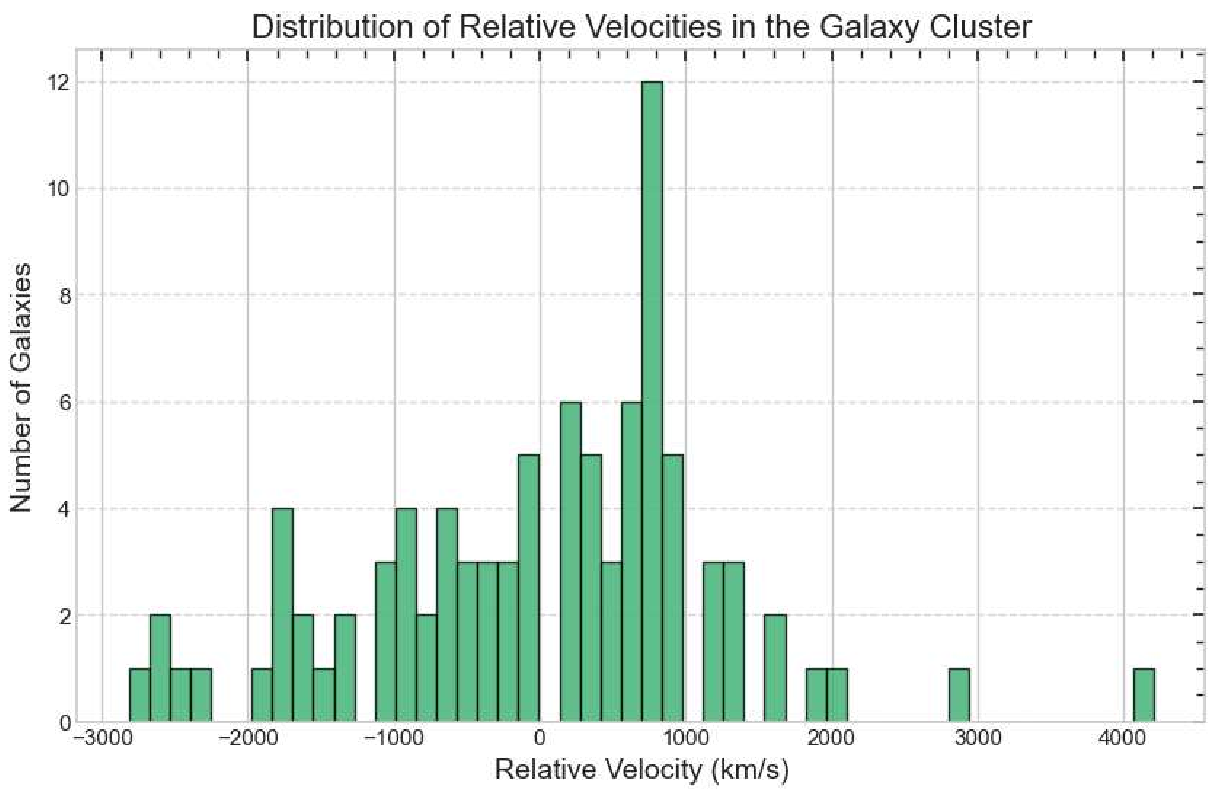
Use the relation between redshift and velocity to add a column named velocity in the data. This would tell the expansion velocity at that redshift

In [23]:
```
c_val = c
G_val = G
```

In [ ]:
```
print("\nAdding Velocity Column")
# Calculate cluster redshift from the filtered data
cluster_redshift = filtered_df['specz'].mean()
print(f"Mean redshift of the identified cluster (from filtered data): {cluster_reds

filtered_df['velocity'] = c_val * (
    ((1 + filtered_df['specz'])**2 - (1 + cluster_redshift)**2) /
    ((1 + filtered_df['specz'])**2 + (1 + cluster_redshift)**2)
)

print("'velocity' column successfully added to 'filtered_df' (values in m/s).")
print("Example velocities (first 5 in km/s):")
print((filtered_df['velocity'].head() / 1000).round(2))
```

```
Adding Velocity Column
Mean redshift of the identified cluster (from filtered data): 0.0801
'velocity' column successfully added to 'filtered_df' (values in m/s).
Example velocities (first 5 in km/s):
0     662.82
1     319.41
2    -139.88
3     214.89
4    1249.41
Name: velocity, dtype: float64
```

In [ ]:
```
#plot the velocity column created as hist
plt.figure(figsize=(10, 6))
plt.hist(filtered_df['velocity'] / 1000, bins=50, color='mediumseagreen', edgecolor
plt.title("Distribution of Relative Velocities in the Galaxy Cluster", fontsize=16)
plt.xlabel("Relative Velocity (km/s)", fontsize=14)
plt.ylabel("Number of Galaxies", fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.minorticks_on()
plt.tick_params(which='both', direction='in', top=True, right=True)
plt.show()
```



In [ ]:
```
#calculate Velocity Dispersion
velocity_dispersion_ms = filtered_df['velocity'].std()

#convert velocity dispersion to km/s for standard astronomical units
velocity_dispersion_kms = velocity_dispersion_ms / 1000

print(f"The characteristic value of velocity dispersion of the cluster along the li
print(f"  = {velocity_dispersion_ms:.2e} m/s")
print(f"  = {velocity_dispersion_kms:.2f} km/s")

#provide some context/sanity check for the calculated dispersion
if velocity_dispersion_kms > 0 and velocity_dispersion_kms < 2000:
    print("\nThis velocity dispersion value is typical for a galaxy cluster.")
elif velocity_dispersion_kms >= 2000:
    print("\nThis velocity dispersion is very high, suggesting a particularly massi
else:
    print("\nThe calculated velocity dispersion is very low or zero. This might ind
    print("data, filtering, or that the selected galaxies do not form a distinct cl
```

```
The characteristic value of velocity dispersion of the cluster along the line of sig
ht (sigma):
  = 1.22e+06 m/s
  = 1219.34 km/s

This velocity dispersion value is typical for a galaxy cluster.
```

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!!

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.

In [29]:
```
mean_cluster_redshift = filtered_df['specz'].mean()
print(f"The mean redshift of the cluster is: {mean_cluster_redshift:.4f}")
```

```
The mean redshift of the cluster is: 0.0801
```

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

In [30]:
```
print(filtered_df['velocity'].describe() / 1000)
```

```
count       0.091000
mean       -2.451027
std      1219.336490
min     -2816.179091
25%      -807.165187
50%       237.343287
75%       755.500236
max      4209.048637
Name: velocity, dtype: float64
```

In [37]:
```
print(f"The value of the cluster redshift = {cluster_redshift:.4}")
print(f"The characteristic value of velocity dispersion of the cluster along the li
```

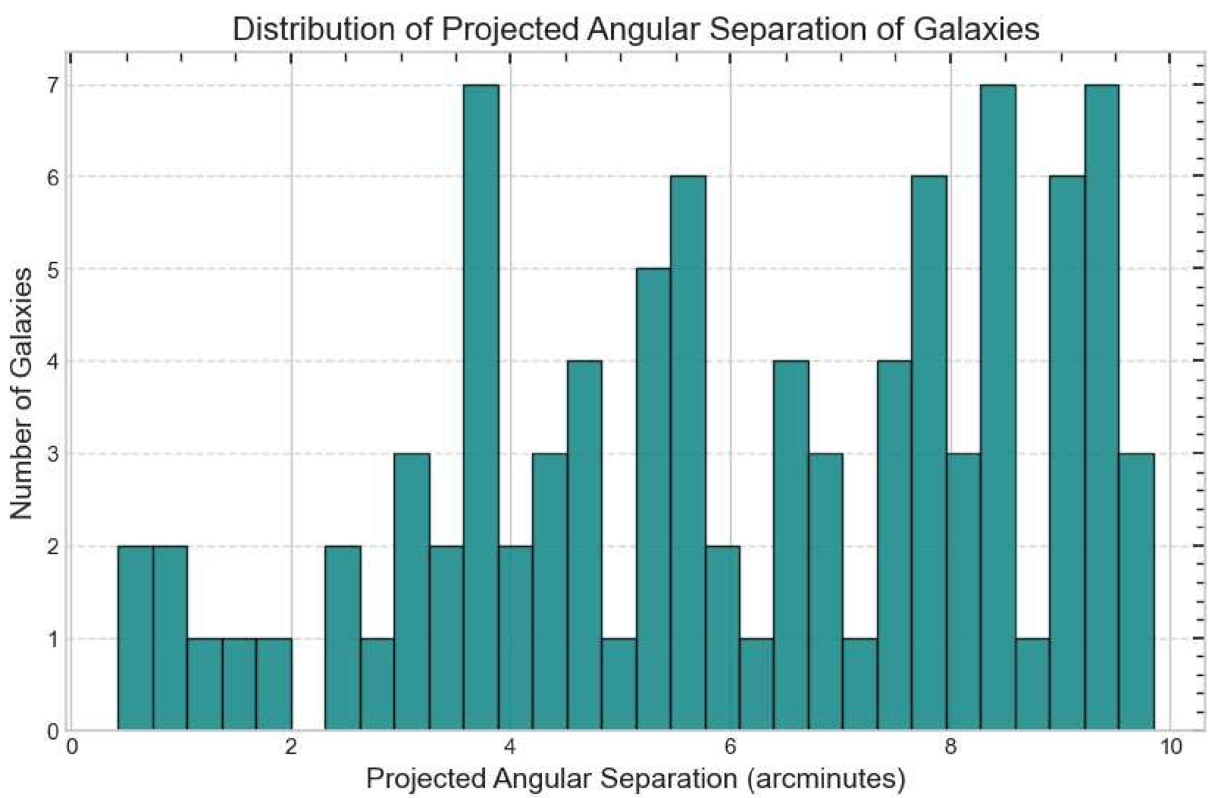
```
The value of the cluster redshift = 0.08007
The characteristic value of velocity dispersion of the cluster along the line of sig
ht = 1.219e+03 km/s.
```

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.

In [32]:
```python
#Plot histogram for proj sep column
plt.figure(figsize=(10, 6))
plt.hist(filtered_df['proj_sep'], bins=30, color='teal', edgecolor='black', alpha=0
plt.title("Distribution of Projected Angular Separation of Galaxies", fontsize=16)
plt.xlabel("Projected Angular Separation (arcminutes)", fontsize=14)
plt.ylabel("Number of Galaxies", fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.minorticks_on()
plt.tick_params(which='both', direction='in', top=True, right=True)
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.

In [33]:
```
r= (c * cluster_redshift / H_0) * (1 - (cluster_redshift / 2) * (1 + q0))
ra= r / (1 + cluster_redshift)

max_proj_sep_arcmin = filtered_df['proj_sep'].max()
# Convert this angular size from arcminutes to radians
angular_size_radians = max_proj_sep_arcmin * (u.arcmin).to(u.rad)
# Calculate physical diameter in meters, then convert to Mpc
diameter_meters = ra * angular_size_radians
diameter = diameter_meters / (3.086e22) # diameter in Mpc (1 Mpc = 3.086e22 meters)

diameter
```

Out[33]:
```
np.float64(0.892837376082904)
```

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 $2 \times 10^{30}\,\mathrm{kg}$.

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

In [38]:
```
### Calculating the dynamical mass in solar masses:
M_dyn =3*((velocity_dispersion_kms*1000)**2)*(diameter*0.5*10**6*3*10**16)/(G*2*10*

print(f"Dynamical Mass of the cluster is {M_dyn:.2e} solar mass")
```

```
Dynamical Mass of the cluster is 4.48e+14 solar mass
```

Project 2: Supernova Cosmology Project

Handbook :

1. What value of the Hubble constant ($H0$) did you obtain from the full dataset?
Answer:
The fitted Hubble constant is approximately 72.97 ± 0.26 km/s/Mpc.

2. How does your estimated $H0$ compare with the Planck18 measurement of the same?
Answer:
Planck18 reports $H_0 \approx 67.4$ km/s/Mpc, so our estimate is higher by ~5.5 km/s/Mpc, which reflects the known Hubble tension between early and late universe measurements.

3. What is the age of the Universe based on your value of H_0? (Assume $\Omega m = 0.3$). How does it change for different values of Ωm?
Answer:

- With $H_0 = 72.97$ and $\Omega_m = 0.3$, the universe's age is about 12.36 billion years.
- Higher Ω_m → slower expansion earlier → younger age.
- Lower Ω_m → faster early expansion → older age.

4. Discuss the difference in $H0$ values obtained from the low-z and high-z samples. What could this imply?
Answer:

- Low-z ($z < 0.1$) gives a higher H_0, while
- High-z ($z \geq 0.1$) gives a lower H_0.
  This suggests possible evolution in cosmic expansion or systematic differences in the data, contributing to the ongoing Hubble tension.

5. Plot the residuals and comment on any trends or anomalies you observe.
Answer:
Residuals are mostly centered around zero, which is expected. However, there may be slight scatter or structure at high redshifts, indicating minor model mismatch or data noise. No large anomalies are visible.

6. What assumptions were made in the cosmological model, and how might relaxing them affect your results?
Answer:

- Assumed a flat ΛCDM model, i.e., flat universe and constant dark energy (Λ).
- Relaxing these (e.g., allowing curvature or evolving dark energy) could change the fit, affect H_0 and Ω_m, and alter the inferred age of the universe.

7. Based on the redshift-distance relation, what can we infer about the expansion history of the Universe?

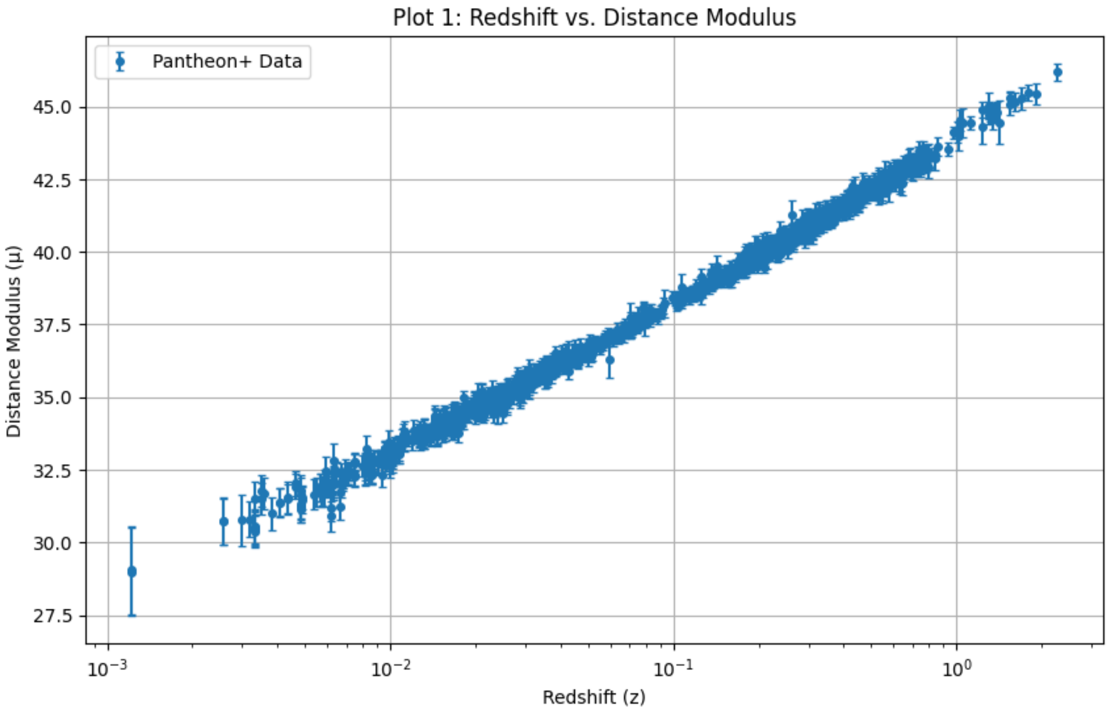
Answer:

It shows that more distant objects (higher z) are receding faster, confirming that the universe is expanding. The shape of the curve also hints at accelerated expansion due to dark energy.

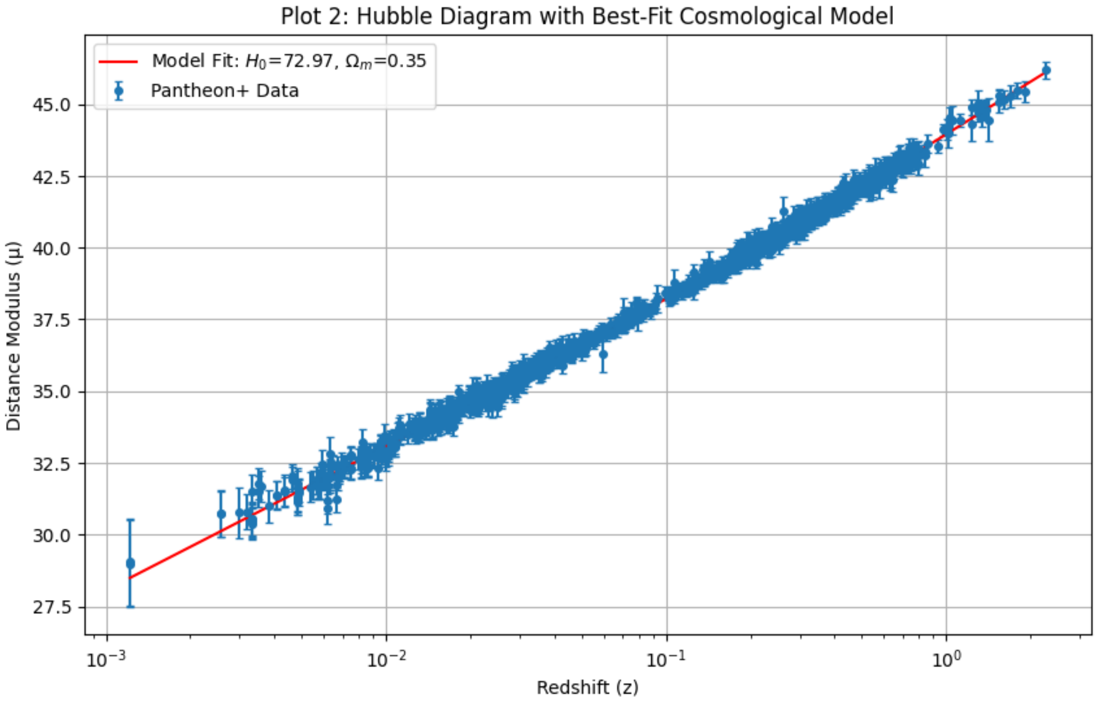
Source Code for this project is uploaded on github to reduce the size of the report for successful project submission.

Github Link - https://github.com/AnshPradhan14/ISA_internship/tree/main/Project_2

Plot 1: Redshift vs. Distance Modulus



Plot 2: Hubble Diagram with Model Fit



Plot 3: Plot the residual