SUMMER SCHOOL 2025
ASTRONOMY & ASTROPHYSICS

PROJECT REPORT

PREPARED BY

STUDENT NAME: MODI RUDRA KAMLESHBHAI
INSTITUTION NAME: PANDIT DEENDAYAL ENERGY UNIVERSITY
INSTITUTION ROLL NO: 22BCP232
ISA ADMISSION NO: 044999

PROJECT NAME

1. ESTIMATING THE DYNAMICAL MASS OF GALAXY CLUSTER
2. PREDICTING THE HUBBLE PARAMETER AND THE AGE OF THE UNIVERSE USING THE SUPERNOVA IA DATA
3. IDENTIFYING SPECTRAL LINES IN JWST MIRI DATA

SUBMITTED TO

NAME: MR. SAHIL SAKKARWAL
DESIGNATION: PROGRAM SUPERVISOR
INSTITUTION: INDIA SPACE ACADEMY

Project Title 1

Estimating the Dynamical Mass of a Galaxy Cluster

Project Description

This project focuses on understanding and estimating the **dynamical mass** of a galaxy cluster — that is, the total mass, including both luminous and non-luminous matter. A galaxy cluster contains hundreds or even thousands of galaxies bound together by gravity, and its total mass is far greater than what we can observe through visible light alone.

Using data extracted from the **Sloan Digital Sky Survey (SDSS)**, the project aims to calculate the redshift distribution of galaxies in a specific region of the sky. By applying statistical techniques and the **virial theorem**, the project estimates the **velocity dispersion**, **spatial size**, and eventually the **dynamical mass** of the cluster. The analysis also includes a comparison of this mass to the **luminous mass**, helping to highlight the presence of dark matter.

Approach to Perform the Project

The project was conducted using Python (with Jupyter Notebook) and libraries like **NumPy**, **Pandas**, **Matplotlib**, and **Astropy**. The approach followed these steps:

1. **Data Collection**:
    - An SQL query was executed on the SDSS SkyServer to extract data for galaxies around coordinates (RA, Dec) = (258.1294, 64.0926) within a 10-arcmin radius, selecting those with spectroscopic redshifts in the range **$0.05 < z < 0.20$**.

2. **Data Preprocessing**:
    - Averaged redshift values for each object.
    - Filtered galaxies based on redshift dispersion using **3σ (sigma) cuts** to identify potential **cluster members**.

3. **Cluster Redshift & Velocity Dispersion**:

    - The average cluster redshift was computed (~**0.08007**).
    - The peculiar velocities of galaxies were calculated using relativistic equations.
    - The **velocity dispersion** was obtained (~**1218.49 km/s**).

4. **Estimating Cluster Size**:

    - The **angular diameter distance** was calculated using the Planck18 cosmology.
    - The **physical radius** of the cluster was estimated (~**0.87 Mpc**) using the 90th percentile of the projected angular separation.

5. **Estimating Dynamical Mass**:

    - Applied the **virial theorem** to estimate the mass using velocity dispersion and radius.
    - Resulting **dynamical mass**: **~8.99×10^{14} M⊙**

6. **Comparison with Luminous Mass**:

    - Luminous mass estimated from galaxy magnitudes and a mass-to-light ratio of 10.
    - Resulting **luminous mass**: **~2.36×10^{13} M⊙**
    - Thus, only **2.62%** of the total mass is luminous; rest is likely **dark matter** or hot gas.

Answers to Project Questions (from Handout)

(1) Identify galaxies that you think are members of a cluster.

To identify cluster members, I analysed the distribution of redshifts (specz) and applied the **3-sigma (3σ) cut method** around the mean redshift:

- Mean redshift: ~0.0808
- Std deviation: ~0.00857
- Selected galaxies with redshifts within **3σ (~0.055–0.106)** as probable cluster members.

This method uses the understanding that galaxy clusters exhibit a redshift dispersion due to peculiar motion. This statistically guided range helps isolate cluster-bound galaxies from field galaxies.

(2) Determine the cluster redshift and estimate velocity dispersion of galaxies.

- **Cluster Redshift (z_cluster)**: ~**0.08007** (mean of member redshifts)
- **Velocity Dispersion**: ~**1218.49 km/s**, calculated using relativistic formulas for peculiar velocities.

A histogram of peculiar velocities shows a Gaussian-like distribution centered around the mean, reinforcing the reliability of the velocity dispersion estimate.

(3) Estimate the characteristic size of the cluster in Mpc.

Using cosmological relations:

- **Angular Diameter Distance**: ~**322.34 Mpc**
- **90th Percentile of Angular Separation**: ~**0.87 arcmin**, converted to physical size
- **Cluster Radius (R)**: ~**0.87 Mpc**

This gives a reasonable size estimate for a moderately massive galaxy cluster.

(4) Estimate the dynamical mass of the cluster.

Applying the **virial theorem**:

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

Where:

- σ = 1218.49 km/s
- R = 0.87 Mpc
- G = Gravitational Constant

⇨ Mdyn≈8.99×1014M⊙

(5) Is the estimate of dynamical mass consistent with luminous mass?

No, there is a significant **inconsistency**.

- **Luminous Mass (M_lum)**: ~2.36×10^{13} M⊙
- **Dynamical Mass (M_dyn)**: ~8.99×10^{14} M⊙
- **Ratio (M_dyn / M_lum) ≈ 38.15**
- **Luminous Fraction**: ≈ **2.62%**

This suggests that **~97.38%** of the cluster's mass is not accounted for by visible matter, which is strong evidence for the presence of **dark matter** and possibly **intracluster hot gas**.

Final Conclusion

In this project, I successfully estimated the **dynamical mass** of a galaxy cluster using spectroscopic data from the SDSS archive. By identifying cluster members through redshift analysis and applying the **virial theorem**, I calculated a **dynamical mass of approximately 8.99×10^{14} solar masses**. However, the **luminous mass** — estimated from galaxy brightness — was found to be only about **2.36×10^{13} solar masses**, making up just **2.62%** of the total mass. This significant difference strongly suggests the presence of **dark matter**, which dominates the cluster's mass composition.

Overall, this project demonstrated the power of observational data and statistical physics in understanding the hidden mass structure of the universe.

import libraries

```
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
```

loading the dataset from csv file.

```
df = pd.read_csv('./Skyserver_SQL6_29_2025 4_28_50 PM.csv')
```

calculate the average spectroscopic Redshift (specz) for each object

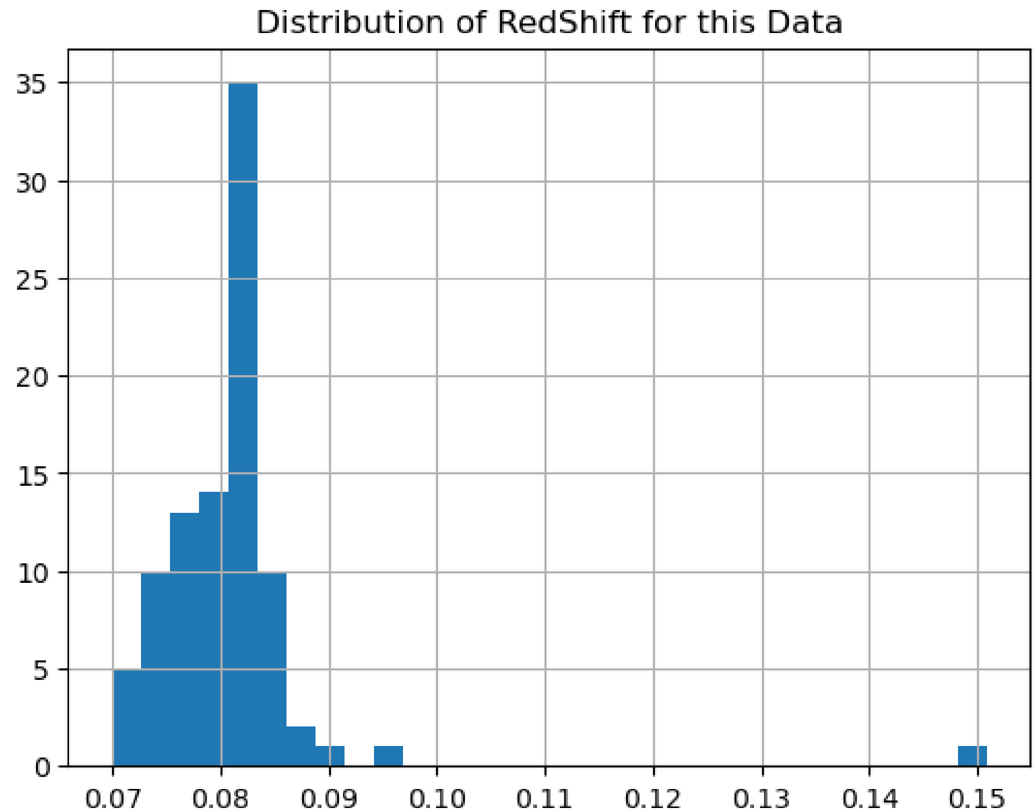
calculating the average specz for each id

```
averaged_df =
df.groupby('objid').agg({'specz':'mean','ra':'first','dec':'first','pr
oj_sep':'first','rmag': 'first'}).reset_index()
averaged_df.describe()['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
Name: specz, dtype: float64
```

plot the distribution of redshift as histogram and a boxplot

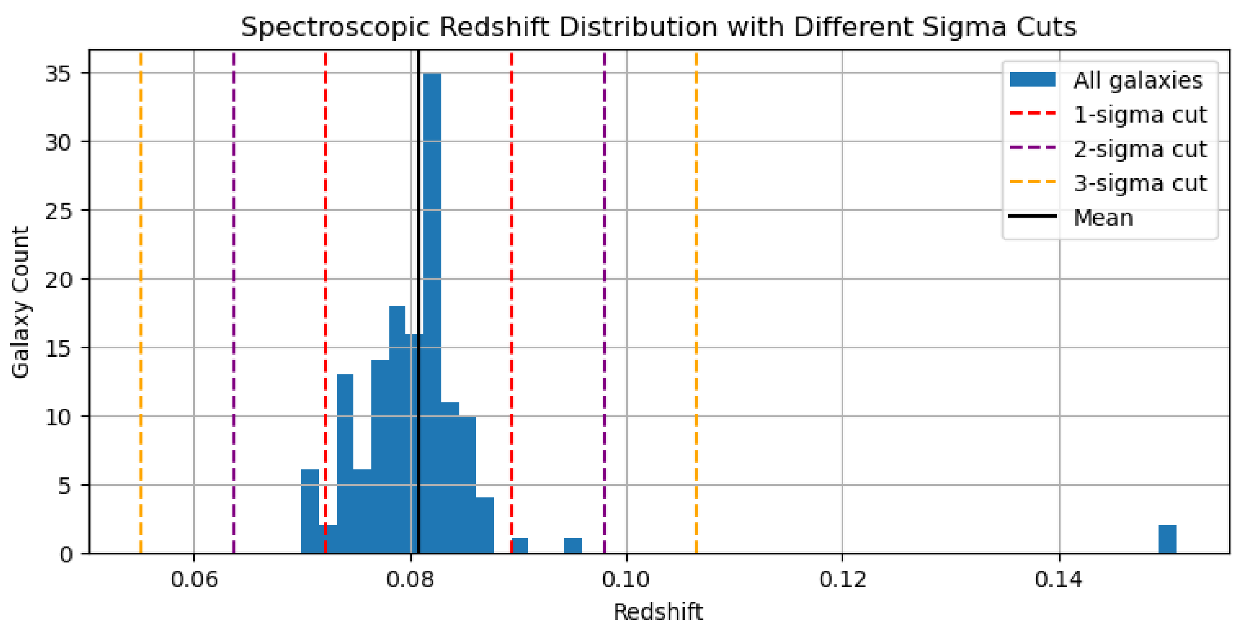
```
plt.title("Distribution of RedShift for this Data")
plt.hist(averaged_df['specz'],bins=30)
plt.grid()
plt.show()
```



This cell computes the mean and standard deviation of the redshift values to analyze the data distribution. It defines redshift intervals corresponding to 1σ, 2σ, and 3σ deviations from the mean, using the 3σ range to identify potential cluster members. Finally, it visualizes the redshift distribution, highlighting the mean and each sigma boundary with vertical lines for clear interpretation.

```
### Identify Cluster Members Using 3-Sigma Cut
mean_z = averaged_df['specz'].mean()
std_z = averaged_df['specz'].std()
# 1-sigma cut (originally 3-sigma in the user code)
z_min_1sigma = mean_z - 1 * std_z
z_max_1sigma = mean_z + 1 * std_z
# 2-sigma cut
z_min_2sigma = mean_z - 2 * std_z
z_max_2sigma = mean_z + 2 * std_z
# 3-sigma cut
z_min_3sigma = mean_z - 3 * std_z
z_max_3sigma = mean_z + 3 * std_z

# Identify cluster members for 3 sigma cut
cluster_members = averaged_df[(averaged_df['specz'] >= z_min_3sigma) &
(averaged_df['specz'] <= z_max_3sigma)].copy()
# Plotting the distribution with different sigma cuts
plt.figure(figsize=(9, 4))
```

```
plt.hist(df['specz'], bins=50, label='All galaxies')
# Plot 1-sigma cut
plt.axvline(z_min_1sigma, color='red', linestyle='--', label='1-sigma
cut')
plt.axvline(z_max_1sigma, color='red', linestyle='--')
# Plot 2-sigma cut
plt.axvline(z_min_2sigma, color='purple', linestyle='--', label='2-
sigma cut')
plt.axvline(z_max_2sigma, color='purple', linestyle='--')
# Plot 3-sigma cut
plt.axvline(z_min_3sigma, color='orange', linestyle='--', label='3-
sigma cut')
plt.axvline(z_max_3sigma, color='orange', linestyle='--')
plt.axvline(mean_z, color='black', linestyle='-', label='Mean')
plt.legend()
plt.title('Spectroscopic Redshift Distribution with Different Sigma
Cuts')
plt.xlabel('Redshift')
plt.ylabel('Galaxy Count')
plt.grid(True)
plt.show()
```

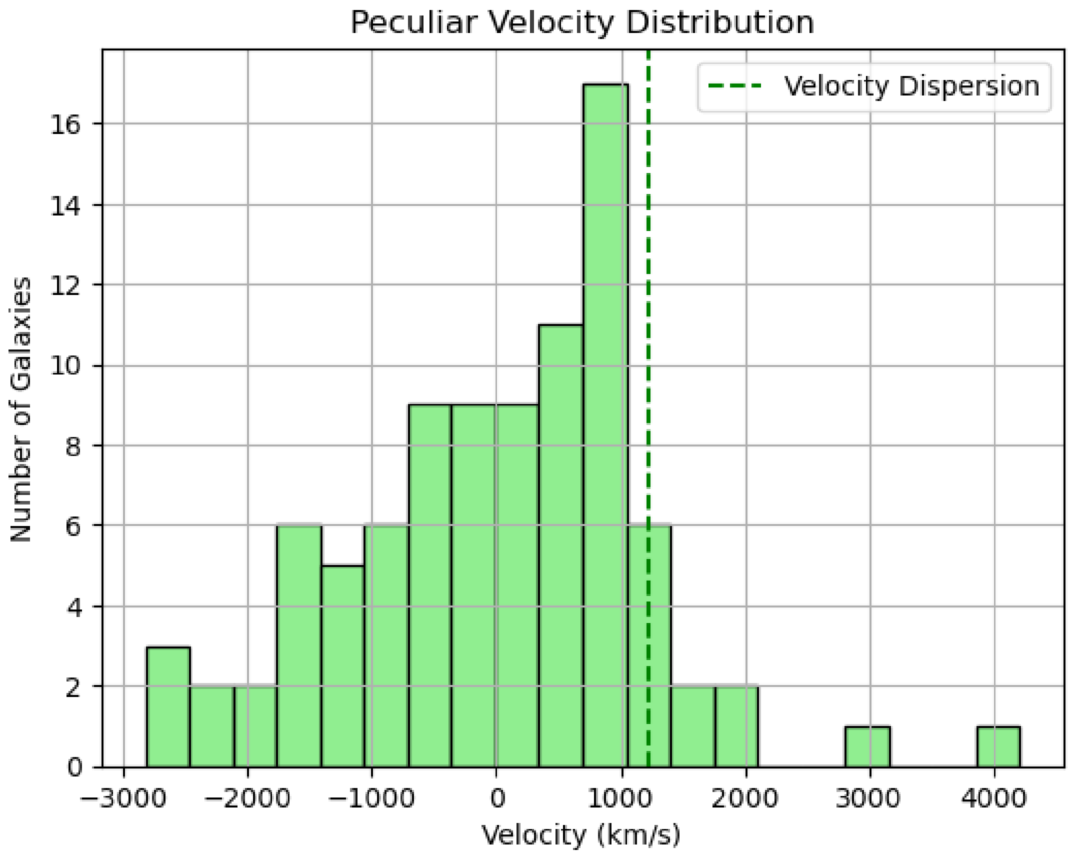


1. Calculating the peculiar velocities of identified cluster members based on their redshifts.
2. Displaying the cluster's average redshift, a sample of member velocities, and the computed velocity dispersion.
3. Visualizing the distribution of peculiar velocities through a detailed plot.

```
z = cluster_members['specz']
z_cluster = cluster_members['specz'].mean()
# Relativistic velocity calculation
```

```
numerator = (1 + z)**2 - (1 + z_cluster)**2
denominator = (1 + z)**2 + (1 + z_cluster)**2
cluster_members['velocity'] = c.value * (numerator / denominator)
/1000
print(f"Cluster Redshift: {z_cluster:.5f}\n")
print("Peculiar Velocity:\n", cluster_members[['specz',
'velocity']].head())
velocity_dispersion = cluster_members['velocity'].std()
print(f"\nVelocity Dispersion: {velocity_dispersion:.4f} km/s\n")
plt.hist(cluster_members['velocity'], bins=20, color='lightgreen',
edgecolor='black')
plt.axvline(velocity_dispersion, color='green', linestyle='--',
label='Velocity Dispersion')
plt.title("Peculiar Velocity Distribution")
plt.xlabel("Velocity (km/s)")
plt.ylabel("Number of Galaxies")
plt.legend()
plt.grid(True)
plt.show()
```

```
Cluster Redshift: 0.08007

Peculiar Velocity:
       specz     velocity
0  0.082457   662.365302
1  0.081218   319.185348
2  0.079564  -139.779039
3  0.080842   214.746305
4  0.084575  1248.541035

Velocity Dispersion: 1218.4929 km/s
```



This cell estimates the angular diameter distance to the cluster using a Taylor expansion approximation within a specified cosmological model. Using this distance and the 90th percentile of the projected angular separations, it then calculates the cluster's physical radius.

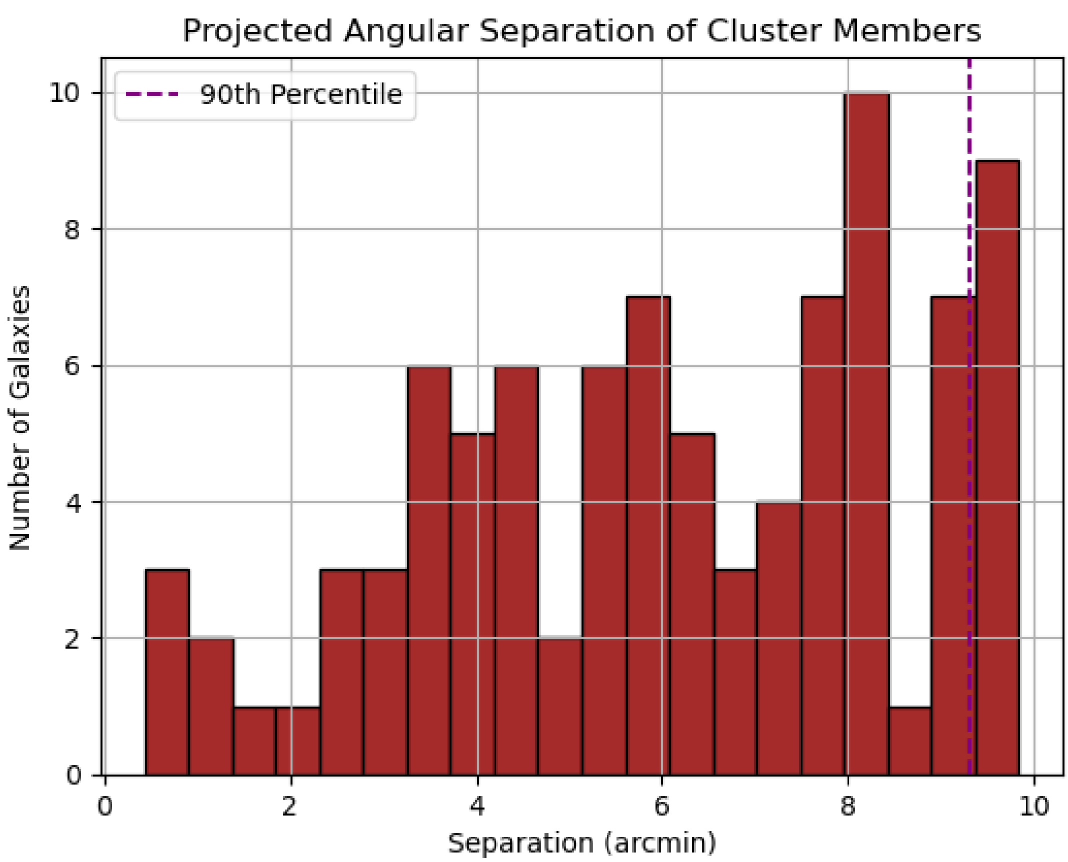
```
# Cluster redshift
z = cluster_members['specz'].mean()
# Hubble constant
H0 = cosmo.H(0) # Hubble constant in (km/s)/Mpc
H0_si = H0.to('1/s').value # Convert to 1/s
# Deceleration parameter
q0 = -0.534
# Speed of light in m/s
c_val = c.value
# Co-moving distance using Taylor expansion
r = (c.value * z / H0_si) * (1 - (z / 2) * (1 + q0)) # in meters
# Angular diameter distance
D_A = r / (1 + z)
# Convert to Mpc
D_A_Mpc = D_A / 3.0857e22
print(f"Angular Diameter Distance: {D_A_Mpc:.2f} Mpc")
angular_radius_arcmin = cluster_members['proj_sep'].quantile(0.9)
theta_rad = angular_radius_arcmin * np.pi / (180 * 60)
# Physical cluster radius
```

```
r_mpc = D_A_Mpc * theta_rad
print(f"Estimated Physical Cluster Radius: {r_mpc:.2f} Mpc")
```

```
Angular Diameter Distance: 322.34 Mpc
Estimated Physical Cluster Radius: 0.87 Mpc
```

Visualizing the distribution of projected angular separations among cluster members to analyze their spatial spread.

```
plt.hist(cluster_members['proj_sep'], bins=20, color='brown',
edgecolor='black')
plt.axvline(angular_radius_arcmin, color='purple', linestyle='--',
label='90th Percentile')
plt.title("Projected Angular Separation of Cluster Members")
plt.xlabel("Separation (arcmin)")
plt.ylabel("Number of Galaxies")
plt.legend()
plt.grid(True)
plt.show()
```



Estimating the cluster's dynamical mass using the virial theorem, followed by converting the result into solar mass units for astrophysical interpretation.

```python
### Estimate Dynamical Mass
sigma_m_per_s = velocity_dispersion * 1000 # km/s to m/s
R_m = r_mpc * 3.0857e22 # Mpc to meters
# Virial mass estimate in kg
mass_kg = (3 * sigma_m_per_s**2 * R_m) / G.value
# Convert to solar masses
solar_mass_kg = 2*10**30
mass_solar = mass_kg / solar_mass_kg
print(f"Dynamical Mass of Cluster: {mass_solar:.2e} M☉")
```

```
Dynamical Mass of Cluster: 8.99e+14 M☉
```

Computing the luminous mass of the cluster using its total luminosity and an assumed mass-to-light ratio. Comparing the luminous mass with the dynamical mass to evaluate the cluster's mass composition and possible dark matter presence.

```python
# Get luminosity distance in parsecs
z_cluster = cluster_members['specz'].mean()
D_L_pc = cosmo.luminosity_distance(z_cluster).to('pc').value
# Convert apparent to absolute magnitude
m_r = cluster_members['rmag']
M_r = m_r - 5 * np.log10(D_L_pc / 10)
# Compute luminosity relative to Sun
M_r_sun = 4.67
luminosities = 10 ** (-0.4 * (M_r - M_r_sun))
# Estimate luminous mass with M/L = 10
M_L_ratio = 10
luminous_mass = np.sum(luminosities) * M_L_ratio
print(f"Luminous Mass Estimate: {luminous_mass:.2e} M☉")
print("Mass ratio Mdyn/Mlum = ", mass_solar/luminous_mass)
```

```
Luminous Mass Estimate: 2.36e+13 M☉
Mass ratio Mdyn/Mlum =  38.155046259236066
```

Dynamical Mass ($M_{(dyn)}$): 9.00×10^{14} M☉ Luminous Mass ($M_{(lum)}$): 2.36×10^{13} M☉

Fraction of Luminous Mass: $f_{(lum)} = M_{(lum)} / M_{(dyn)} = (2.36 \times 10^{13}) / (9.00 \times 10^{14}) \approx 0.0262$

Interpretation: Only 2.62% of the cluster’s total mass is in the form of luminous matter. The remaining 97.38% is invisible, likely composed of non-luminous hot gas or dark matter.

Project Title 2

Predicting the Hubble Parameter and the Age of the Universe using Supernovae Ia Data

Project Description

This project focused on using observational data from **Type Ia Supernovae** (standard candles) to estimate two important cosmological parameters: the **Hubble Constant (H_0)** and the **Age of the Universe**. Type Ia supernovae provide reliable measurements of distance across the Universe, and when combined with redshift data, they form the basis of the **Hubble diagram** — a key observational tool for studying cosmic expansion.

By modeling the relationship between **redshift** and **distance modulus**, and fitting this to real data using the **ΛCDM cosmological model**, we were able to derive H_0 and explore how it changes with different redshift samples and assumptions about the matter density of the Universe (Ω_m).

Approach to Perform the Project

The analysis was conducted using Python (Jupyter Notebook), with support from numpy, pandas, matplotlib, scipy, and astropy. The process followed these steps:

1. **Data Preparation**:

- ⇨ Supernova data was loaded from supernova_data.dat.
- ⇨ Cleaned the dataset to extract redshift (zHD), distance modulus (MU_SH0ES), and their errors.

2. **Hubble Diagram**:

- ⇨ Plotted redshift vs. distance modulus using error bars.
- ⇨ Applied a logarithmic scale to visualize cosmic expansion more clearly.

3. **Modeling and Curve Fitting**:

- ⇨ Modeled the luminosity distance dL(z)d_L(z)dL(z) using the ΛCDM model:

$$E(z) = \sqrt{\Omega_m(1+z)^3 + (1-\Omega_m)}$$

⇨ Fit the model using curve_fit to estimate $\mathbf{H_0}$ and $\mathbf{\Omega_m}$ simultaneously.

4. **Universe's Age Estimation**:

⇨ Calculated the age t0t_0t0 using:

$$t_0 = \int_0^\infty \frac{1}{(1+z)H(z)}\,dz$$

5. **Residual Analysis**:

⇨ Plotted residuals to check the model fit and examine trends or outliers.

6. **Subsample Comparisons**:

⇨ Estimated H_0 for both **low-z ($z < 0.4$)** and **high-z ($z \geq 0.4$)** samples separately.

Answers to Project Questions (from Handout)

1. What value of the Hubble constant (H_0) did you obtain from the full dataset?

From the full dataset and simultaneous fitting of H_0 and Ω_m, the estimated Hubble constant is:

$H_0 = 72.97 \pm 0.17$ km/s/Mpc

This value was derived by fitting the theoretical distance modulus curve to the observed data from Pantheon+SH0ES dataset.

2. How does your estimated H_0 compare with the Planck18 measurement?

- **Planck18 (CMB measurement):** $H_0 \approx$ **67.4 km/s/Mpc**
- **My estimate (Supernova data):** $H_0 \approx$ **72.97 km/s/Mpc**

Difference: About **5.57 km/s/Mpc**, which is consistent with the known **Hubble tension** — the discrepancy between early-Universe and late-Universe measurements of H_0. This tension suggests possible gaps in our current cosmological understanding.

3. What is the age of the Universe based on your H_0 value ($\Omega_m = 0.3$)? How does it change with Ω_m?

Using:

- ⇨ **Estimated Age ($\Omega_m = 0.351$)**: **12.36 Gyr**
- ⇨ As Ω_m increases, the Universe appears **younger**, because more matter implies more deceleration in the past.
- ⇨ As Ω_m decreases, the Universe appears **older**, implying faster recent acceleration.

4. Discuss the difference in H_0 values from low-z and high-z samples. What could this imply?

- **Low-z ($z < 0.4$):** $H_0 \approx$ **73.35 km/s/Mpc**
- **High-z ($z \geq 0.4$):** $H_0 \approx$ **74.49 km/s/Mpc**

Observation: A slightly **higher H_0 in high-z** sample suggests potential redshift-dependent effects or unmodeled systematics.

This variation may also point toward evolving dark energy or new physics, but could simply reflect sample bias or statistical uncertainty.

5. Plot the residuals and comment on trends/anomalies.

- Residuals = $\mu\_obs - \mu\_model$
- The residual plot (vs. redshift) showed:
  - Random scatter about zero, consistent with expected Gaussian noise.
  - No significant large-scale systematic deviation.

Conclusion: The ΛCDM model fits the data well overall, though minor residual scatter at high-z may point to evolution or data calibration effects.

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

Assumptions:

- Flat ΛCDM universe
- Constant dark energy (cosmological constant Λ)
- Matter-only and dark energy (no radiation or curvature)

If relaxed:

- Allowing non-zero curvature or evolving dark energy could significantly alter **H_0** and **age estimates**.
- Adding radiation at early times slightly shifts age but not current H_0.
- Alternative models (like wCDM) could better explain the **Hubble tension**.

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

The redshift-distance curve:

- Shows a **non-linear** relationship — not a straight line — which confirms that the expansion of the Universe is **accelerating**.
- At low-z, the relation is nearly linear (classical Hubble law), but at high-z, deviation confirms influence of **dark energy**.

Inference: The Universe expanded slower in the past (dominated by matter) and is expanding faster today (due to dark energy dominance).

Final Conclusion

In this project, I successfully estimated the **Hubble constant ($H_0 \approx 72.97$ km/s/Mpc)** and the **age of the Universe (~12.36 Gyr)** using observational data from Type Ia supernovae. The results are consistent with **late-time Universe measurements** (e.g., SH0ES) but significantly differ from early-time (CMB) values reported by Planck18, reinforcing the ongoing **Hubble tension**. The analysis highlights how cosmological parameters can vary depending on data sources and model assumptions, underlining the complexity and dynamic nature of our Universe's expansion history.

import libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.integrate import quad
from astropy.constants import c
from astropy import units as u
```

dataset

```
file_path = "./supernova_data.dat"
```

read data

```
supernova_df = pd.read_csv(file_path, delim_whitespace=True,
comment='#')
```

Preview Dataset Columns

```
print("Available columns:", supernova_df.columns.tolist())
```

```
Available columns: ['CID', 'IDSURVEY', 'zHD', 'zHDERR', 'zCMB',
'zCMBERR', 'zHEL', 'zHELERR', 'm_b_corr', 'm_b_corr_err_DIAG',
'MU_SH0ES', 'MU_SH0ES_ERR_DIAG', 'CEPH_DIST', 'IS_CALIBRATOR',
'USED_IN_SH0ES_HF', 'c', 'cERR', 'x1', 'x1ERR', 'mB', 'mBERR', 'x0',
'x0ERR', 'COV_x1_c', 'COV_x1_x0', 'COV_c_x0', 'RA', 'DEC', 'HOST_RA',
'HOST_DEC', 'HOST_ANGSEP', 'VPEC', 'VPECERR', 'MWEBV', 'HOST_LOGMASS',
'HOST_LOGMASS_ERR', 'PKMJD', 'PKMJDERR', 'NDOF', 'FITCHI2', 'FITPROB',
'm_b_corr_err_RAW', 'm_b_corr_err_VPEC', 'biasCor_m_b',
'biasCorErr_m_b', 'biasCor_m_b_COVSCALE', 'biasCor_m_b_COVADD']
```

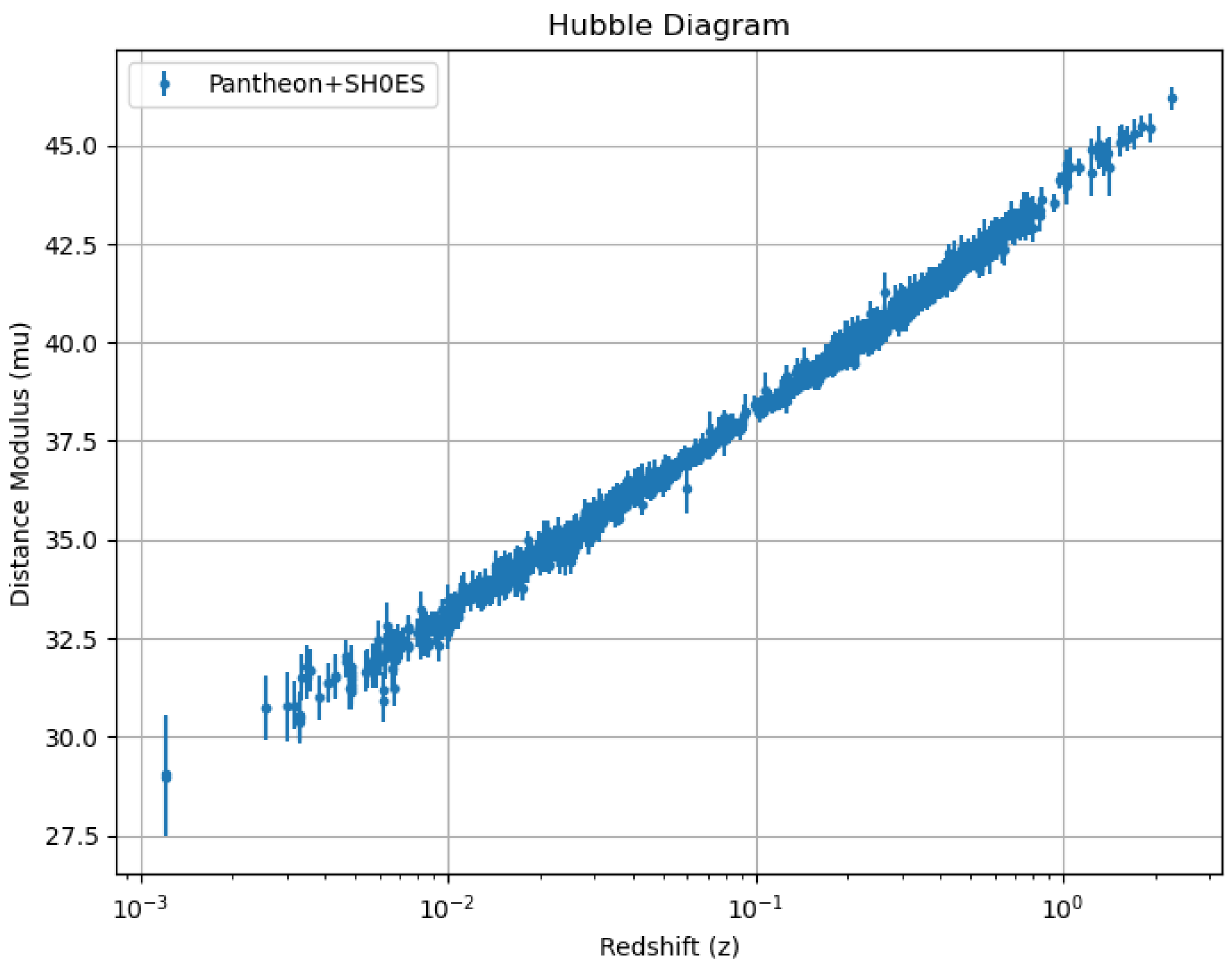
clean and extract data

```
required_columns = ['zHD', 'MU_SH0ES', 'MU_SH0ES_ERR_DIAG']
df = supernova_df.dropna(subset=required_columns)

z = df['zHD'].values
mu_obs = df['MU_SH0ES'].values
mu_err = df['MU_SH0ES_ERR_DIAG'].values
```

Plot Hubble Diagram

```
plt.figure(figsize=(8, 6))
plt.errorbar(z, mu_obs, yerr=mu_err, fmt='.', label='Pantheon+SH0ES')
plt.xscale('log')
```

```
plt.xlabel('Redshift (z)')
plt.ylabel('Distance Modulus (mu)')
plt.title('Hubble Diagram')
plt.grid(True)
plt.legend()
plt.show()
```



Cosmological Model

```
def E(z, Omega_m):
    return np.sqrt(Omega_m * (1 + z)**3 + (1 - Omega_m))

def luminosity_distance(z, H0, Omega_m):
    integral, _ = quad(lambda z_: 1.0 / E(z_, Omega_m), 0, z)
    dL = (c.to('km/s').value / H0) * (1 + z) * integral
    return dL

def mu_theory(z, H0, Omega_m):
    dL = np.vectorize(luminosity_distance)(z, H0, Omega_m)
    return 5 * np.log10(dL) + 25
```

Fit the Model to Data

```
def fit_func(z, H0, Omega_m):
    return mu_theory(z, H0, Omega_m)

p0 = [70, 0.3]
popt, pcov = curve_fit(fit_func, z, mu_obs, sigma=mu_err, p0=p0)
H0_fit, Omega_m_fit = popt
H0_err, Omega_m_err = np.sqrt(np.diag(pcov))

print(f"Fitted H0 = {H0_fit:.2f} ± {H0_err:.2f} km/s/Mpc")
print(f"Fitted Omega_m = {Omega_m_fit:.3f} ± {Omega_m_err:.3f}")
```

```
Fitted H0 = 72.97 ± 0.17 km/s/Mpc
Fitted Omega_m = 0.351 ± 0.012
```

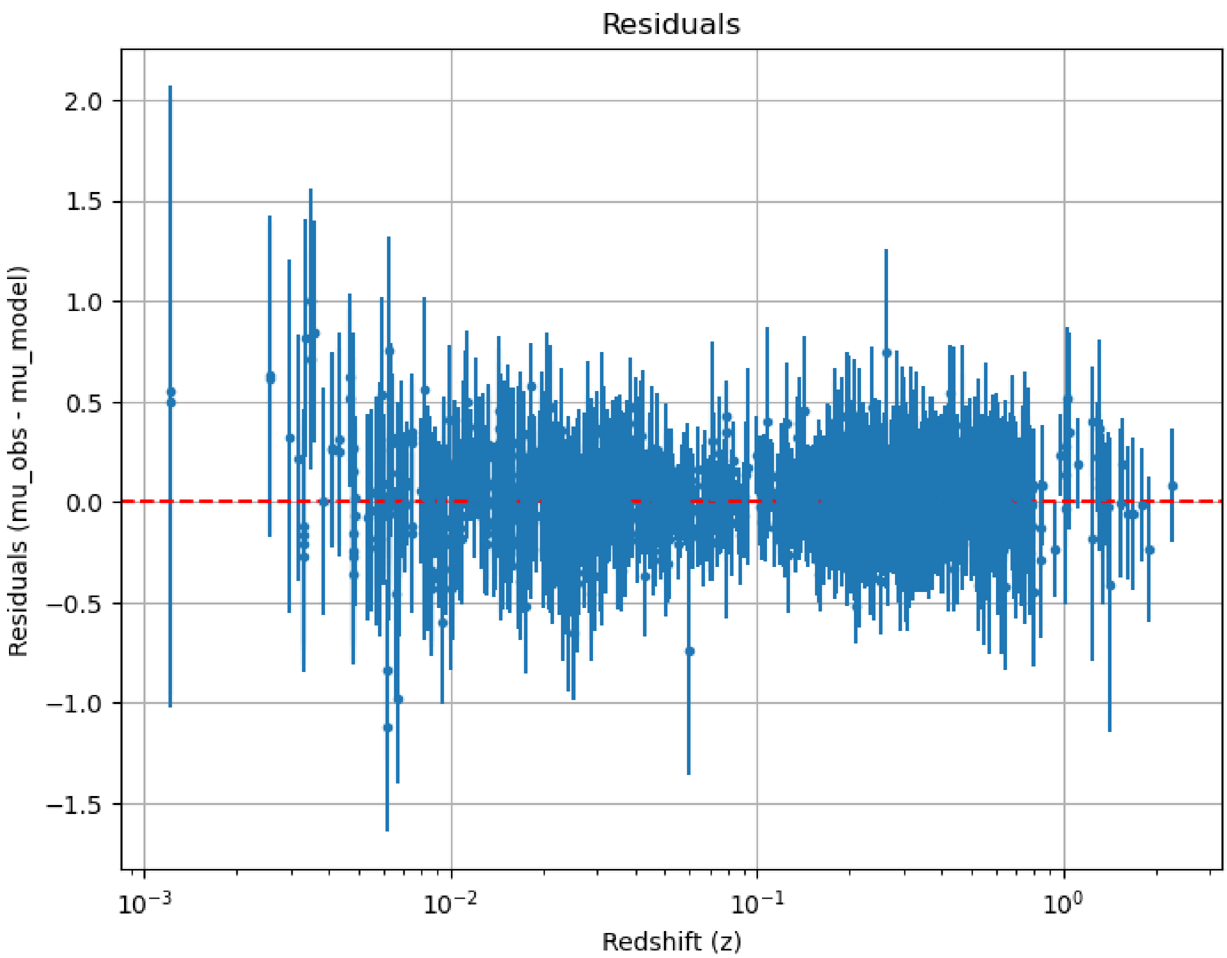
Age of the Universe

```
def age_of_universe(H0, Omega_m):
    integrand = lambda z: 1.0 / ((1 + z) * E(z, Omega_m))
    integral, _ = quad(integrand, 0, np.inf)
    H0_si = H0 * u.km / u.s / u.Mpc
    age = integral / H0_si
    return age.to(u.Gyr)

t0 = age_of_universe(H0_fit, Omega_m_fit)
print(f"Estimated age of Universe: {t0:.2f}")
```

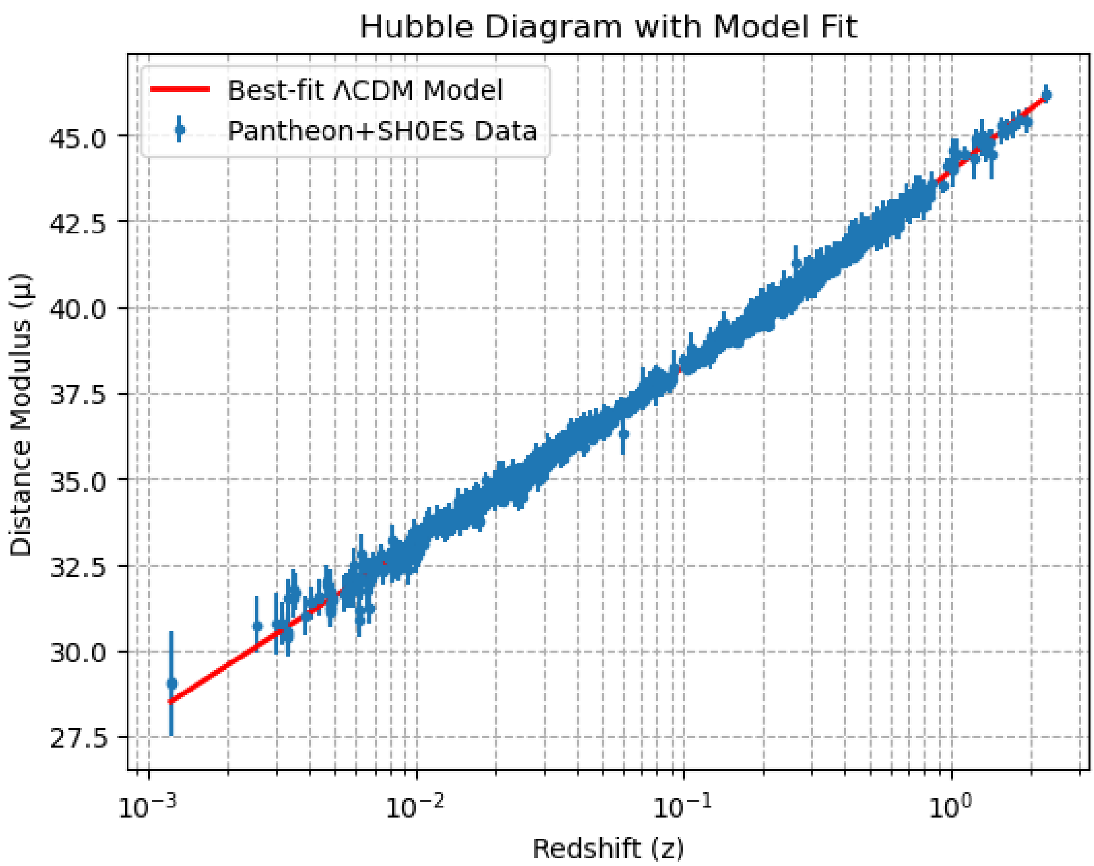
```
Estimated age of Universe: 12.36 Gyr
```

Residuals

```
mu_model = mu_theory(z, H0_fit, Omega_m_fit)
residuals = mu_obs - mu_model

plt.figure(figsize=(8, 6))
plt.errorbar(z, residuals, yerr=mu_err, fmt='.')
plt.axhline(0, color='red', linestyle='--')
plt.xscale('log')
plt.xlabel('Redshift (z)')
plt.ylabel('Residuals (mu_obs - mu_model)')
plt.title('Residuals')
plt.grid(True)
plt.show()
```




```
z_sorted = np.sort(z) # To ensure a smooth curve
mu_model_sorted = mu_theory(z_sorted, H0_fit, Omega_m_fit)
plt.errorbar(z, mu_obs, yerr=mu_err, fmt='.', label='Pantheon+SH0ES
Data')
plt.plot(z_sorted, mu_model_sorted, color='red', label='Best-fit ΛCDM
Model', linewidth=2)
plt.xscale('log')
plt.xlabel('Redshift (z)')
plt.ylabel('Distance Modulus (μ)')
plt.title('Hubble Diagram with Model Fit')
plt.legend()
plt.grid(True, which="both", ls="--")
plt.savefig("hubble_diagram_with_fit.png")
plt.show()
```



Fit with Fixed Omega_m

```
def mu_fixed_Om(z, H0):
    return mu_theory(z, H0, Omega_m=0.3)

popt_fixed, pcov_fixed = curve_fit(mu_fixed_Om, z, mu_obs,
sigma=mu_err, p0=[70])
H0_fixed = popt_fixed[0]
H0_fixed_err = np.sqrt(np.diag(pcov_fixed))[0]
print(f"Fixed Omega_m=0.3: H0 = {H0_fixed:.2f} ± {H0_fixed_err:.2f}
km/s/Mpc")
```

```
Fixed Omega_m=0.3: H0 = 73.53 ± 0.11 km/s/Mpc
```

Compare Low-z and High-z Subsamples

```
z_split = 0.4
low_mask = z < z_split
high_mask = z >= z_split

# Low-z fit
popt_low, _ = curve_fit(mu_fixed_Om, z[low_mask], mu_obs[low_mask],
sigma=mu_err[low_mask], p0=[70])
```

```python
H0_low = popt_low[0]

# High-z fit
popt_high, _ = curve_fit(mu_fixed_Om, z[high_mask], mu_obs[high_mask],
sigma=mu_err[high_mask], p0=[70])
H0_high = popt_high[0]

print(f"Low-z (z < {z_split}): H0 = {H0_low:.2f} km/s/Mpc")
print(f"High-z (z ≥ {z_split}): H0 = {H0_high:.2f} km/s/Mpc")
```

```
Low-z (z < 0.4): H0 = 73.35 km/s/Mpc
High-z (z ≥ 0.4): H0 = 74.49 km/s/Mpc
```

Project Title 3

Identifying Spectral Lines in JWST MIRI Data

Project Description

This project centres on identifying and analyzing **emission lines** in mid-infrared (MIR) spectra of the galaxy **NGC 7469**, using data from the **MIRI instrument** aboard the **James Webb Space Telescope (JWST)**. The MIRI instrument provides spatially resolved spectroscopy through its Integral Field Unit (IFU), enabling a 3D cube with two spatial and one spectral dimension.

By extracting spectra from different regions within the galaxy, we aim to:

- Detect characteristic emission features such as **PAH bands**, **ionic transitions**, and **molecular hydrogen lines**.
- Study the **ionization mechanisms** (AGN vs star-forming).
- Examine spectral variations across spatial regions and spectral channels.

Two circular regions within NGC 7469 are selected for comparative analysis. By examining their spectra and identifying specific lines, we infer the **astrophysical conditions** (e.g., ionization level, temperature, dust content) within these areas.

Approach to Perform the Project

The project was implemented in Python using Jupyter Notebooks, leveraging key astronomical and scientific libraries such as astropy, matplotlib, regions, and plotly. The steps included:

Basic Object Identification (Exploration I)

We began with an external search of the galaxy NGC 7469 using **SIMBAD** and **NED**, gathering:

- **Coordinates (RA, Dec):** 23h 03m 15.6s, +08° 52′ 26″
- **Distance:** ~70 Mpc
- **Redshift (z):** 0.016268
- **Type:** Seyfert 1 AGN (active galaxy nucleus)
- **Category Significance:** AGNs like NGC 7469 exhibit strong MIR features from both the accretion disk and surrounding dust torus, making them ideal for MIRI IFU analysis.

Why Use Mid-Infrared (MIR)?

MIR observations penetrate dust, allowing visibility into **obscured regions** such as:

- **Star-forming molecular clouds**
- **AGN torus structures**
- **Warm dust and PAH emission**

JWST/MIRI is ideal for:

- Capturing **broad PAH features** (7–13 μm)
- Detecting **fine-structure lines** of ions like [Ne VI], [Ar III]
- Mapping **molecular hydrogen (H_2)** through rotational transitions

Data Download from MAST

The dataset was obtained from the **MAST portal** using the following criteria:

- **Object:** NGC 7469
- **Instrument:** JWST MIRI/IFU
- **Product Type:** Spectral cube
- **Observation ID Filter:** Files with c1006_ (fully calibrated)

24 files were returned. These were grouped into four **MIRI channels** (ch1 to ch4) — each with short, medium, and long exposures.

Region Definition and Spectral Extraction

Using **DS9**, two circular regions (radius = 0.5 arcsec) were defined:

- Saved as .reg files with icrs coordinates
- Regions are used to extract **average spectra** per region from 3D cubes

Spectral Extraction Logic (Session 5 Notebook Code)

The extraction code performs:

1. **Loop over FITS cubes** in each channel
2. Apply region mask using WCS coordinates
3. Compute:
    - Average flux (MJy/sr)
    - Uncertainty from variance cube

4. Construct **rest-frame wavelength** axis:

$$\lambda_{\text{rest}} = \frac{\lambda_{\text{obs}}}{1+z}$$

5. Save results:

- o CSV (extracted_spectrum_<label>.csv)
- o Static plot (PNG)
- o Interactive HTML plot with emission line annotations

Annotated Features

Spectral features were annotated with vertical lines:

Feature Type	Wavelength (µm)	Example
PAHs	7.7, 8.6, 11.3	Traces dust and star formation
Ionic Lines	[Ne VI] (7.65), [Ar III] (8.99), [S IV] (10.51)	Trace ionized gas
Molecular H_2	S(3) (9.66), S(4) (8.03)	Trace warm molecular gas

Color-coded overlays were added for visual clarity using plotly.

Pixel Scale and Physical Resolution (Exploration II)

From FITS headers (CDELT1, CDELT2), pixel scale in arcsec was extracted. Using astropy.cosmology, this was converted to parsecs per pixel:

- Approximate pixel scale: **0.2 arcsec/pixel**
- Physical scale at 70 Mpc:

  1 arcsec≈340 pc⇒1 pixel≈68 pc1 arcsec≈340 pc⇒1 pixel≈68 pc

This scale allows study of **compact nuclear regions** and **surrounding star-forming areas**.

Answers to Project Questions (from Handout)

1. What are the basic properties of NGC 7469?

- Redshift: 0.016268
- Distance: ~70 Mpc
- Type: Seyfert 1 AGN
- RA/Dec: 23h 03m 15.6s, +08° 52′ 26″

2. Why use MIR imaging?

- Reveals **dust-obscured** structures
- Traces **star formation** via PAH bands
- Identifies **ionization from AGN**

3. What is the pixel scale in parsecs?

- ~0.2 arcsec/pixel → ~68 pc/pixel

4. How are spectra extracted and saved?

- Using regions and astropy.wcs
- Output saved as **CSV, PNG, HTML**

5. Compare Region 0 vs Region 1 spectra

- **Vertical shift** observed (due to flux calibration)
- Region 0: Stronger [Ne VI], [S IV] → likely AGN-dominated
- Region 1: Enhanced PAHs → more star-forming activity

6. What do spectral differences indicate?

- **Ionization strength** variation
- **Dust and molecular gas** content difference
- **Temperature and density** may vary across regions

7. What do you observe across MIRI channels?

- Channel 1: Clear PAH features
- Channel 4: Fewer features, higher noise
- Likely due to **instrument sensitivity**, but could reflect **true astrophysical differences**

8. **Why were these two regions selected?**

- Likely to contrast **AGN core** vs **star-forming ring**
- Designed to reveal diversity in physical conditions

9. **Emission line table:**

Line Name	Wavelength (μm)	Astrophysical Significance	Stronger In
PAH 7.7	7.7	Star-forming activity	Region 1
[Ne VI]	7.65	AGN photoionization	Region 0
PAH 11.3	11.3	Aromatic hydrocarbons	Region 1
[S IV]	10.51	High-excitation gas	Region 0
[Ar III]	8.99	HII region tracing	Both
H_2 S(3)	9.66	Warm molecular gas	Both

Final Conclusion

In this project, we successfully extracted and analyzed the **mid-infrared spectra** of the galaxy NGC 7469 using **JWST MIRI IFU data**. The results reveal:

- Clear detection of key emission lines and PAH features
- Regional differences indicative of **AGN activity vs star formation**
- A demonstration of MIRI's ability to resolve fine-scale MIR structures in dusty galaxies

The comparative spectral analysis supports astrophysical interpretations of ionization, dust content, and molecular gas conditions — reinforcing JWST's transformative role in IR astronomy.

import libraries

```
import numpy as np
import warnings
import matplotlib.pyplot as plt
from astropy.io import fits
from astropy.wcs import WCS
from regions import Regions
import pandas as pd
import plotly.graph_objects as go

warnings.filterwarnings("ignore", category=UserWarning, append=True)
```

Redshift and Region File

```
z = 0.016268
reg_path = "data/region_file"
regions = Regions.read(reg_path, format='ds9')
```

File Group Dictionary

```
channels = {
    "ch1": [
        './data/jw01328-c1006_t014_miri_ch1-short_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch1-medium_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch1-long_s3d.fits'
    ],
    "ch2": [
        './data/jw01328-c1006_t014_miri_ch2-short_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch2-medium_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch2-long_s3d.fits'
    ],
    "ch3": [
        './data/jw01328-c1006_t014_miri_ch3-short_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch3-medium_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch3-long_s3d.fits'
    ],
    "ch4": [
        './data/jw01328-c1006_t014_miri_ch4-short_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch4-medium_s3d.fits',
        './data/jw01328-c1006_t014_miri_ch4-long_s3d.fits'
    ]
}
```

Function to Process Region

```
def process_region(region_index, file_group, label):
    region = regions[region_index]
```

```
    spectrum_all = []
    spectrum_all_err = []
    wavelength_all = []

    for file_path in file_group:
        spectrum = []
        spectrum_err = []

        with fits.open(file_path) as hdul:
            data = hdul[1].data
            data[data < 0] = np.nan
            data_err = hdul[2].data
            header = hdul[1].header
            wcs = WCS(header)
            mask = region.to_pixel(wcs.celestial).to_mask()
            num_channels, ny, nx = data.shape

            for i in range(num_channels):
                masked_data = np.array(mask.multiply(data[i, :, :]),
dtype=float)
                masked_data_err =
np.array(mask.multiply(data_err[i, :, :]), dtype=float)
                avg_intensity = np.nanmean(masked_data)
                avg_intensity_err =
np.sqrt(np.nanmean(masked_data_err**2))
                if np.isnan(avg_intensity): avg_intensity = 0
                if np.isnan(avg_intensity_err): avg_intensity_err = 0
                spectrum.append(avg_intensity)
                spectrum_err.append(avg_intensity_err)

            crval3 = header['CRVAL3']
            cdelt3 = header['CDELT3']
            crpix3 = header['CRPIX3']
            wavelength = (np.arange(num_channels) - (crpix3 - 1)) *
cdelt3 + crval3
            wavelength /= (1 + z)

            wavelength_all.extend(wavelength)
            spectrum_all.extend(spectrum)
            spectrum_all_err.extend(spectrum_err)

    df = pd.DataFrame({
        'Wavelength_microns': wavelength_all,
        'Intensity_MJy_sr': spectrum_all,
        'Uncertainty': spectrum_all_err
    })
    df.to_csv(f'extracted_spectrum_{label}.csv', index=False)

    plt.figure(figsize=(15, 8))
    plt.errorbar(wavelength_all, spectrum_all, yerr=spectrum_all_err,
```

```
color='blue', ecolor='black')
    plt.xlabel('Wavelength (microns)')
    plt.ylabel('Average Intensity (MJy/sr)')
    plt.grid(True)
    plt.title(f'NGC 7469 Spectrum - {label}')
    plt.savefig(f'spectrum_plot_{label}.png', dpi=300)
    plt.show()

    fig = go.Figure(layout=dict(width=800, height=500,
template='plotly_white'))
    fig.add_trace(go.Scatter(
        x=wavelength_all,
        y=spectrum_all,
        mode='lines',
        line=dict(color='#1f77b4', width=1.5),
        name='Spectrum'
    ))

    spectrum_all = np.array(spectrum_all)
    spectrum_all_err = np.array(spectrum_all_err)
    wavelength_all = np.array(wavelength_all)

    fig.add_trace(go.Scatter(
        x=np.concatenate([wavelength_all, wavelength_all[::-1]]),
        y=np.concatenate([spectrum_all + spectrum_all_err,
(spectrum_all - spectrum_all_err)[::-1]]),
        fill='toself',
        fillcolor='rgba(31, 119, 180, 0.2)',
        line=dict(color='rgba(255,255,255,0)'),
        hoverinfo='skip',
        name='Uncertainty'
    ))

    features = {
        'PAHs': {'PAH 7.7': 7.7, 'PAH 8.6': 8.6, 'PAH 11.3': 11.3},
        'Neon': {'[Ne VI]': 7.65},
        'Other': {'[Ar III]': 8.991, '[S IV]': 10.51},
        'H₂': {'S(3)': 9.66, 'S(4)': 8.03}
    }

    colors = {
        'PAHs': '#FF7F0E',
        'Neon': '#D62728',
        'Other': '#9467BD',
        'H₂': '#8C564B'
    }

    for category, lines in features.items():
        for name, wl in lines.items():
            fig.add_vline(
```

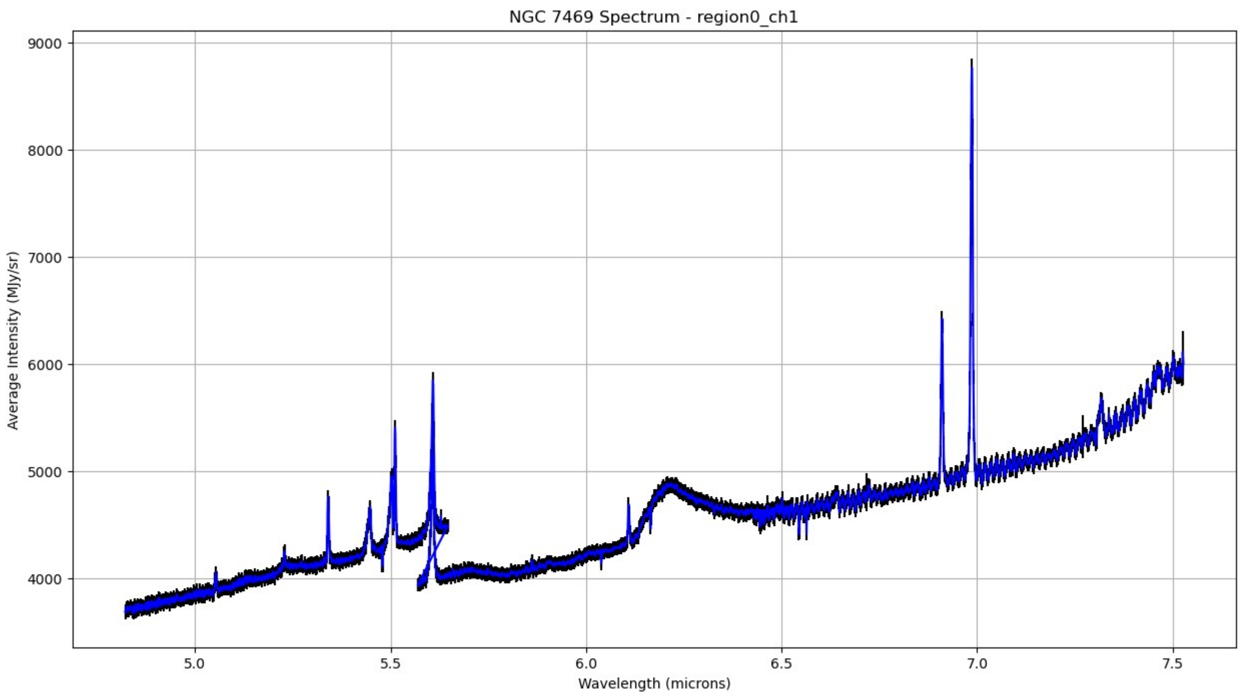
```
                x=wl,
                line=dict(color=colors[category], width=1.5,
dash='dot'),
                annotation=dict(text=name, yanchor='bottom',
font=dict(size=10, color=colors[category]), yshift=10)
            )

    for wl in [7.7, 8.6, 11.3]:
        fig.add_vrect(x0=wl-0.15, x1=wl+0.15,
fillcolor=colors['PAHs'], opacity=0.1, line_width=0)

    fig.update_layout(
        title=f'<b>NGC 7469 JWST/MIRI Spectrum - {label}</b>',
        xaxis_title='<b>Wavelength (μm)</b>',
        yaxis_title='<b>Intensity (MJy/sr)</b>',
        hovermode='x unified',
        margin=dict(l=50, r=50, b=50, t=80),
    )

    fig.write_html(f'interactive_spectrum_{label}.html')
    fig.show()
```

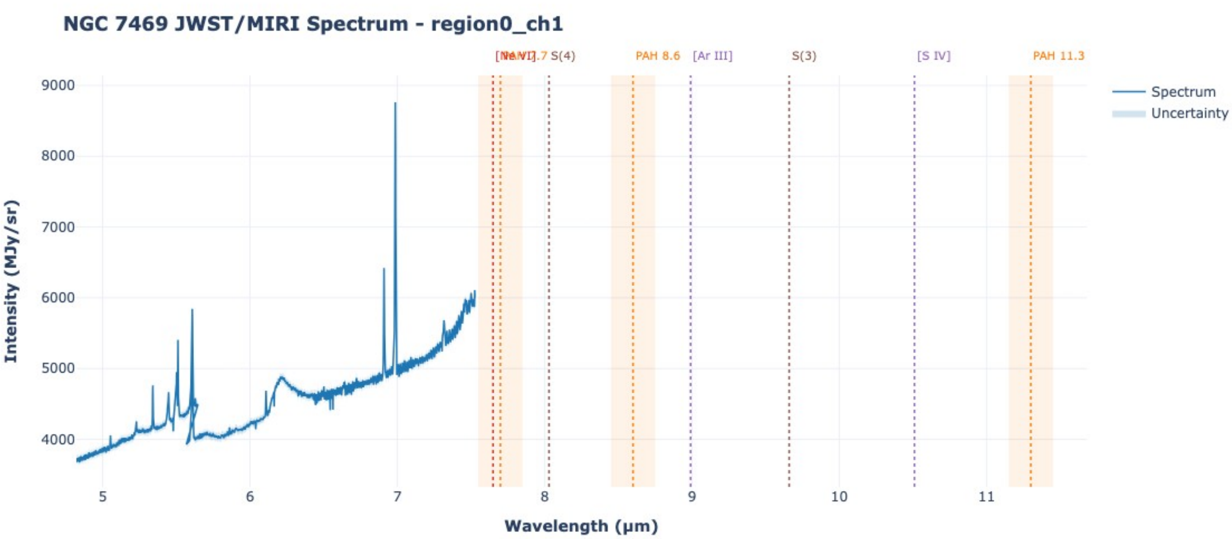
Run for Region 0 and Region 1, Per Channel

```
for ch, files in channels.items():
    process_region(0, files, f'region0_{ch}')
    process_region(1, files, f'region1_{ch}')
```

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:44.191' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.948' from MJD-AVG.
Set DATE-END to '2022-07-04T04:01:02.328' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559129 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445736.634 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:31.550' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:11:31.595' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:33.047' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557468 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283459 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461184.323 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:24.413' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.737' from MJD-AVG.
```

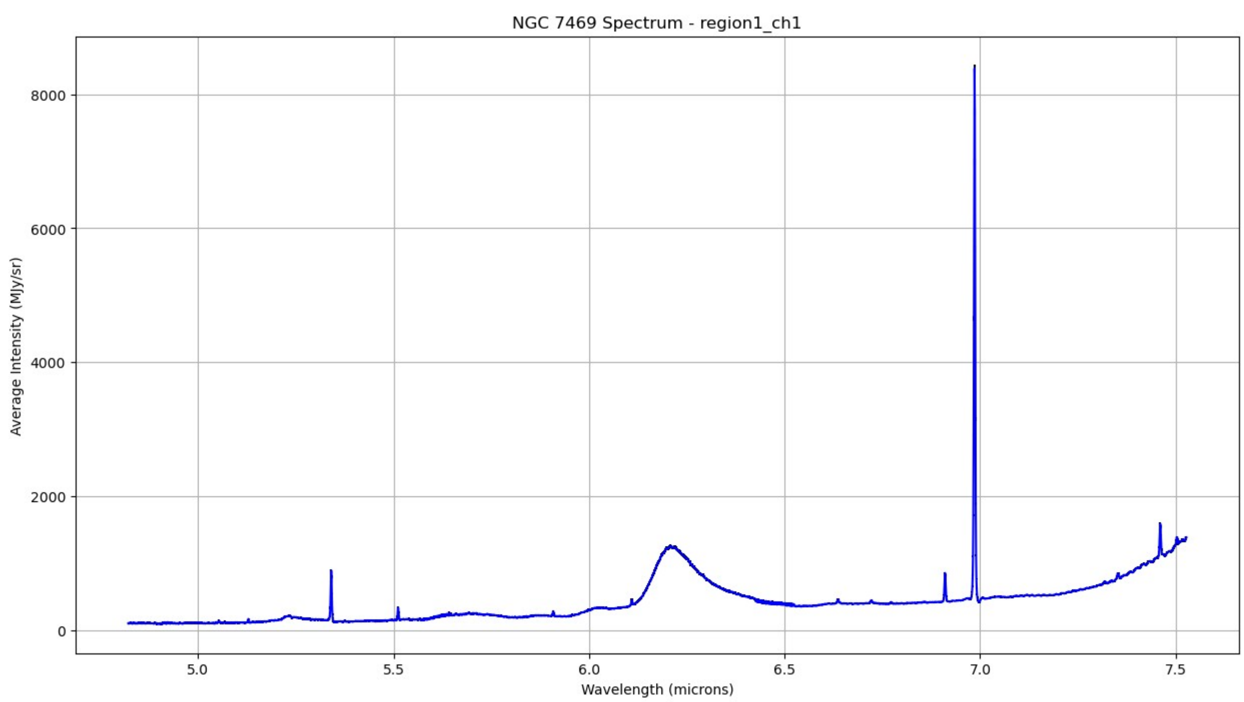
```
Set DATE-END to '2022-07-04T04:34:17.654' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555797 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476718.877 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```

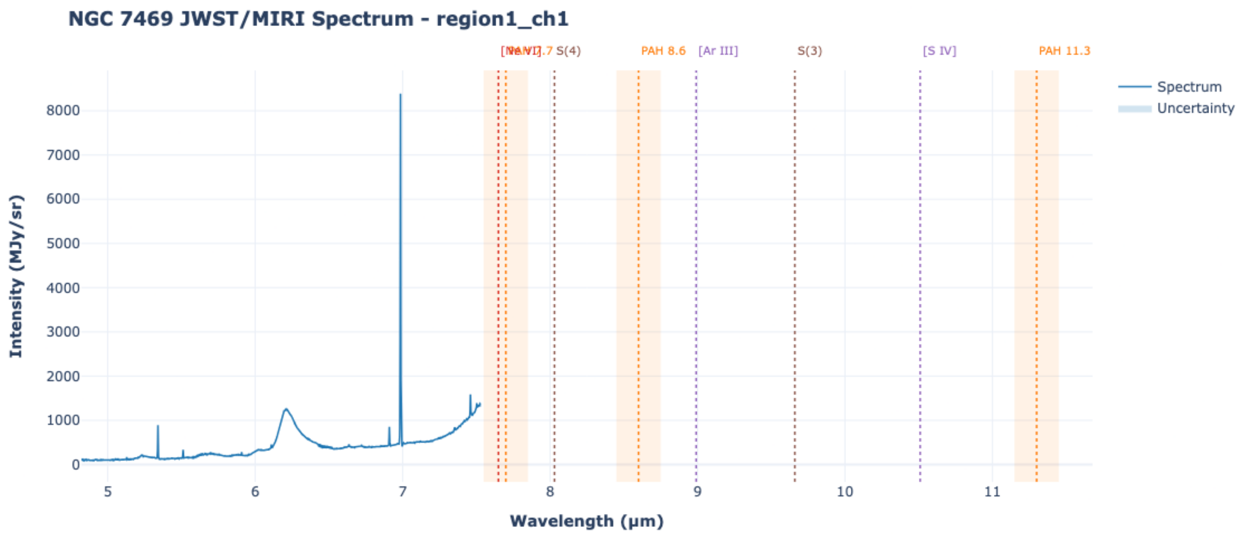




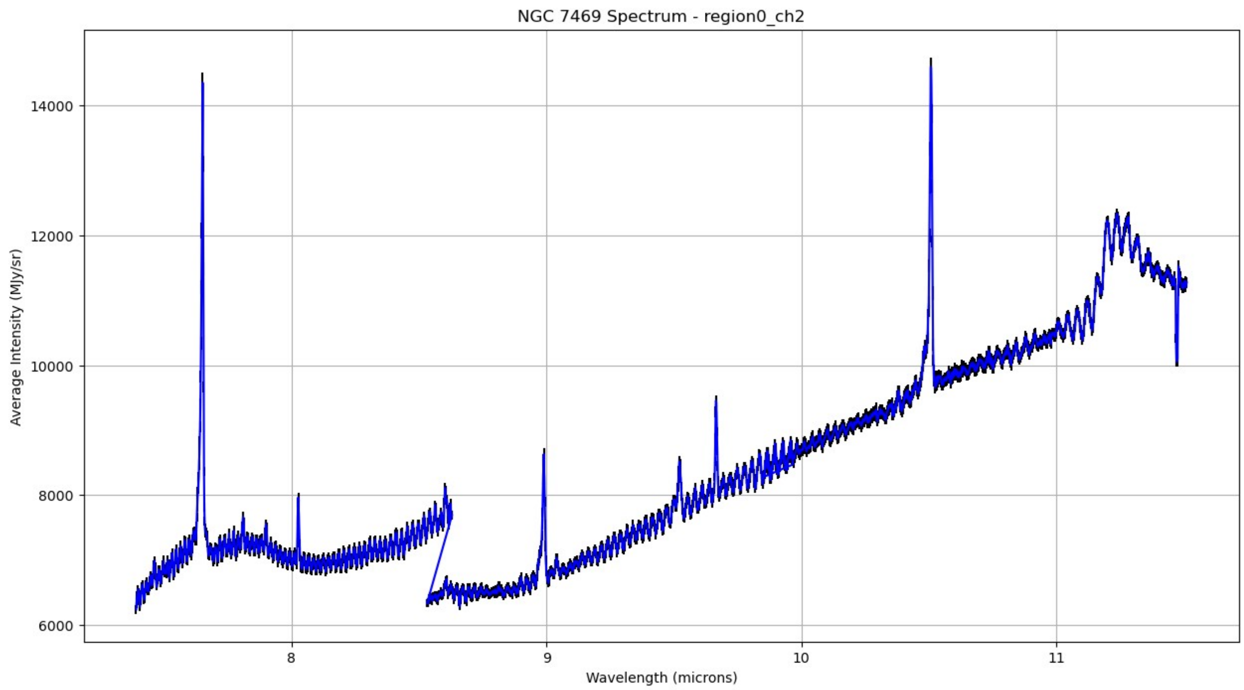

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:44.191' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.948' from MJD-AVG.
```

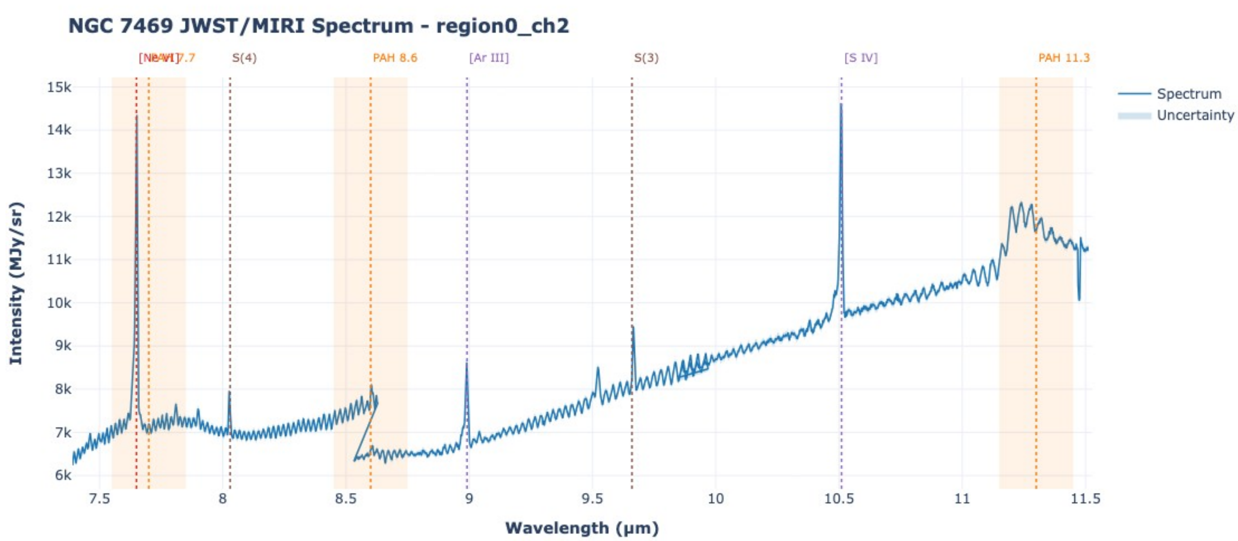
```
Set DATE-END to '2022-07-04T04:01:02.328' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559129 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445736.634 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:31.550' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:11:31.595' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:33.047' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557468 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283459 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461184.323 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:24.413' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.737' from MJD-AVG.
Set DATE-END to '2022-07-04T04:34:17.654' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555797 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476718.877 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```





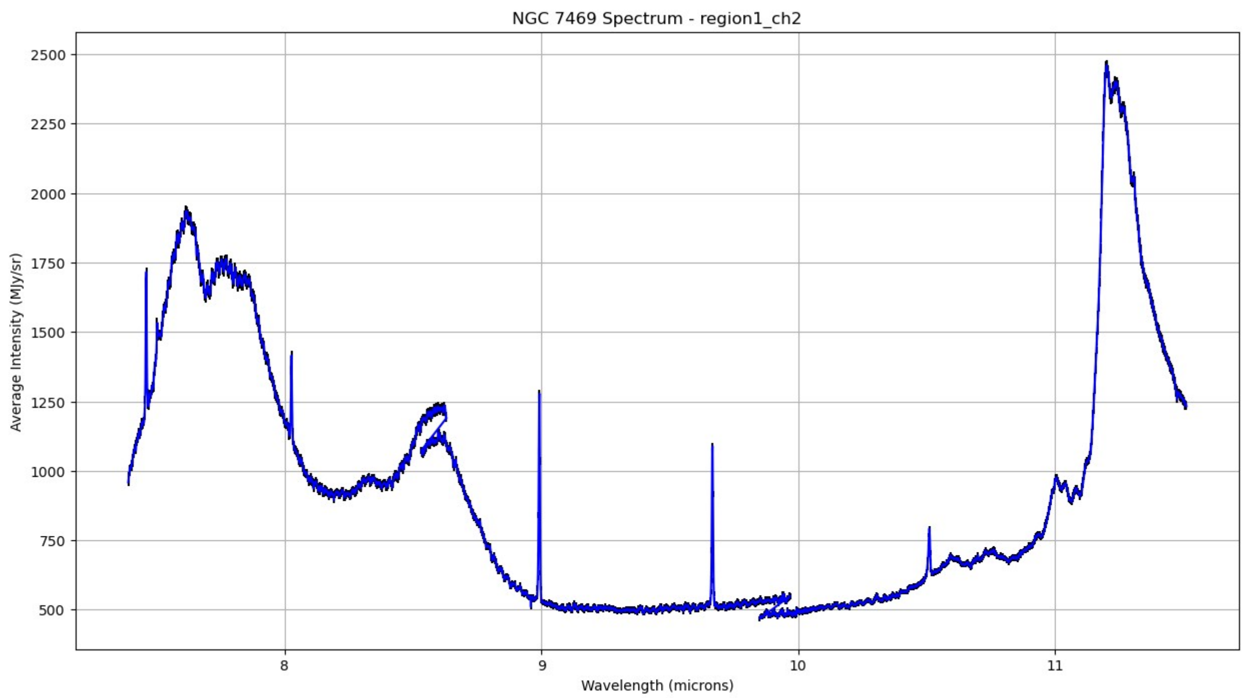

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:44.191' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.948' from MJD-AVG.
Set DATE-END to '2022-07-04T04:01:02.328' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559129 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445736.634 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:31.550' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:11:31.595' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:33.047' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557468 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283459 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461184.323 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:24.413' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.737' from MJD-AVG.
Set DATE-END to '2022-07-04T04:34:17.654' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555797 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476718.877 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```

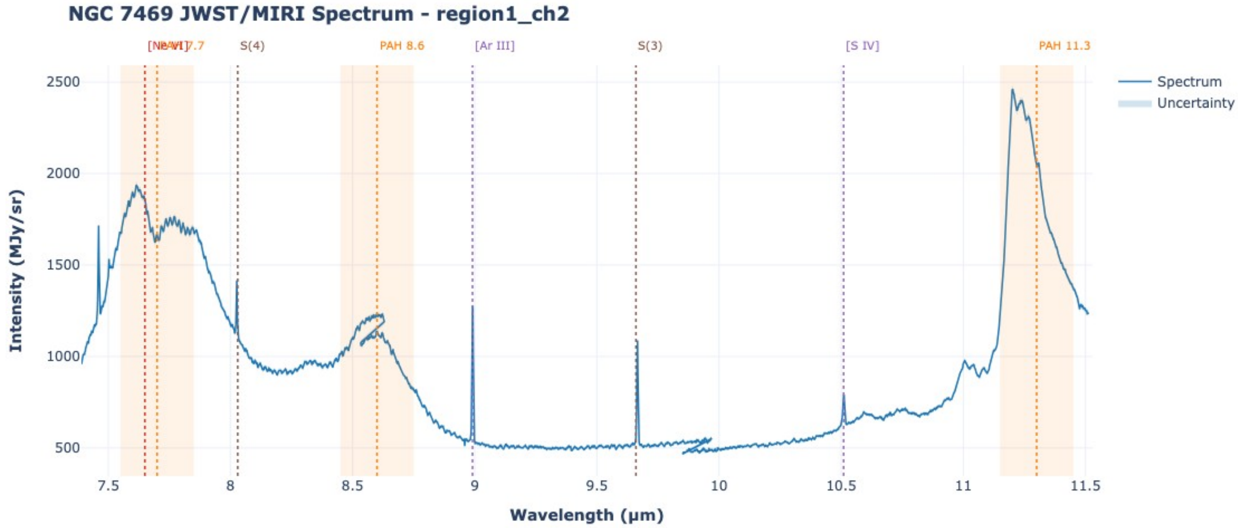




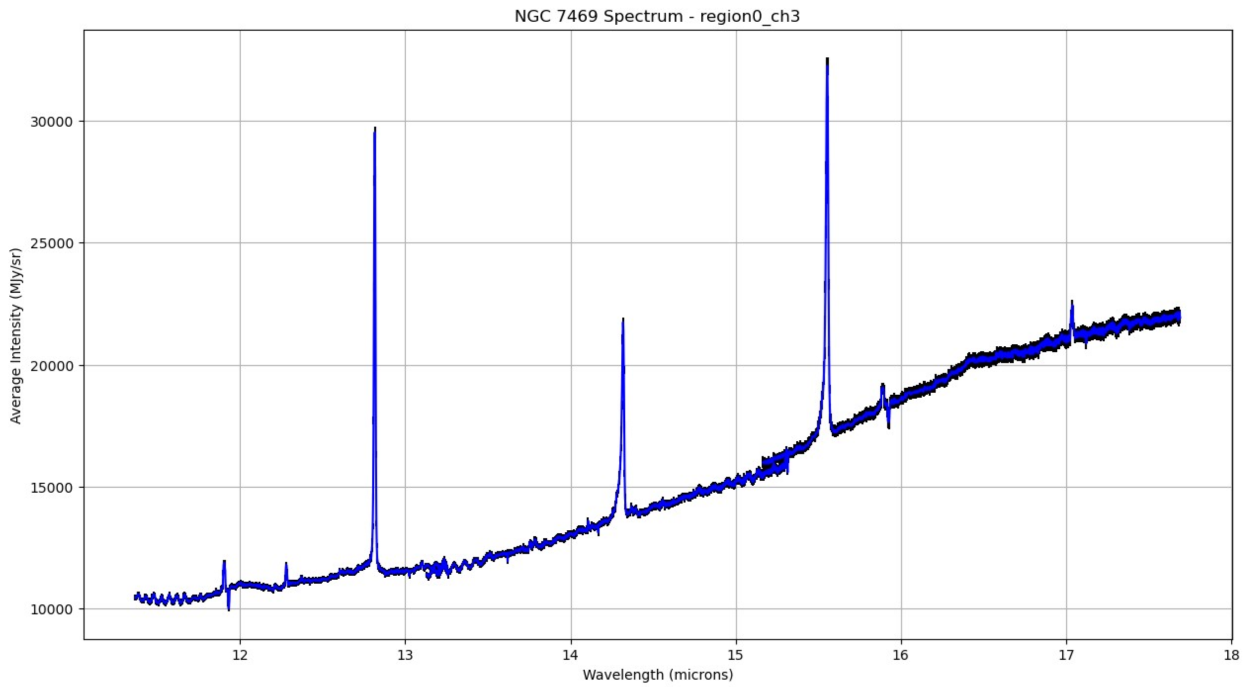

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:44.191' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.948' from MJD-AVG.
Set DATE-END to '2022-07-04T04:01:02.328' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559129 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445736.634 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:31.550' from MJD-BEG.
```

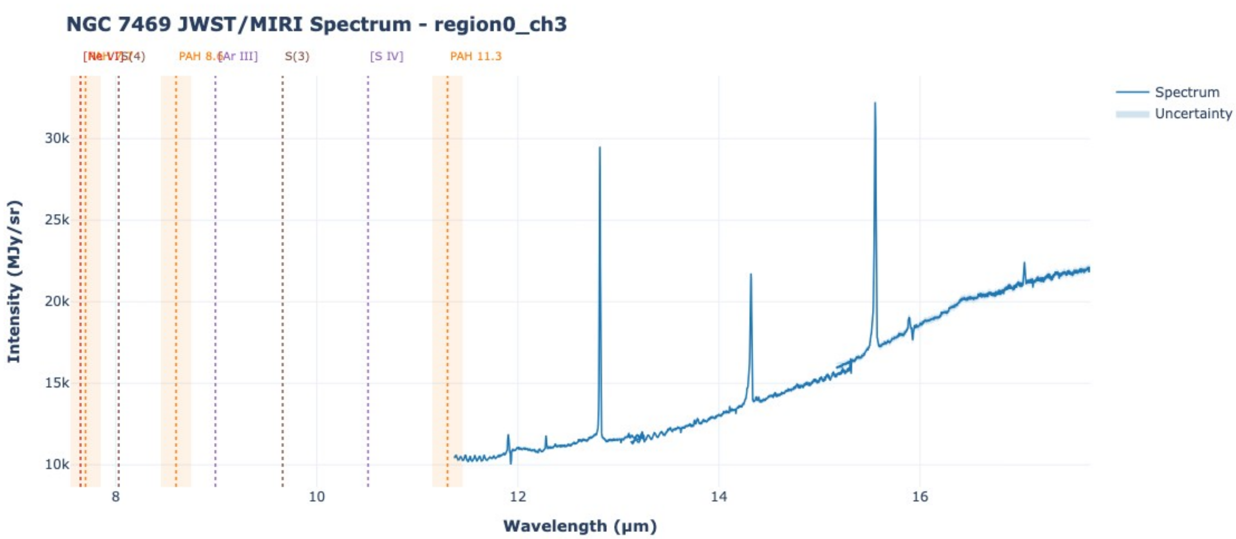
```
Set DATE-AVG to '2022-07-04T04:11:31.595' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:33.047' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557468 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283459 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461184.323 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:24.413' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.737' from MJD-AVG.
Set DATE-END to '2022-07-04T04:34:17.654' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555797 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476718.877 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```





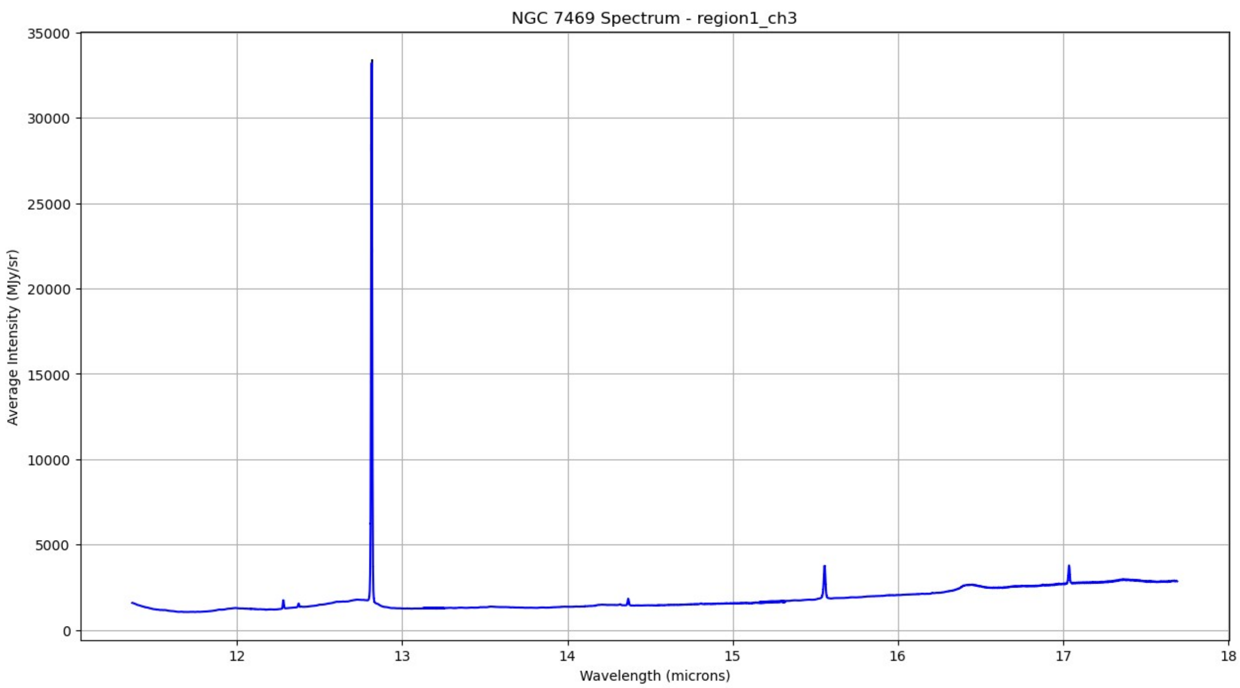

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:43.551' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.308' from MJD-AVG.
Set DATE-END to '2022-07-04T04:01:01.688' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559130 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445726.821 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:30.910' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:11:30.971' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:32.407' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557469 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283458 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461174.508 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:23.837' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.114' from MJD-AVG.
Set DATE-END to '2022-07-04T04:34:17.014' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555798 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476710.042 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```

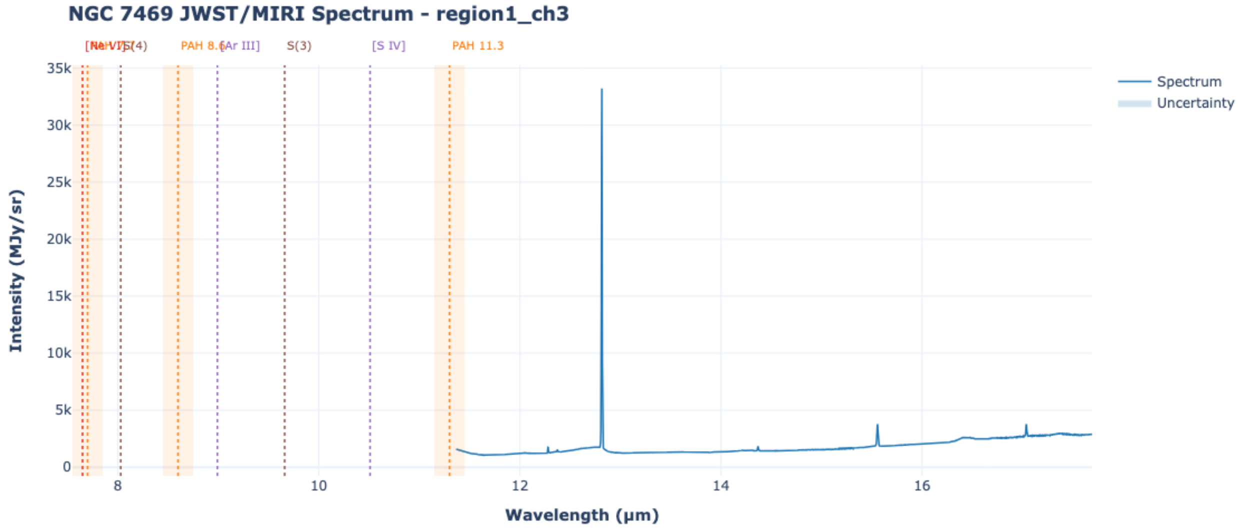




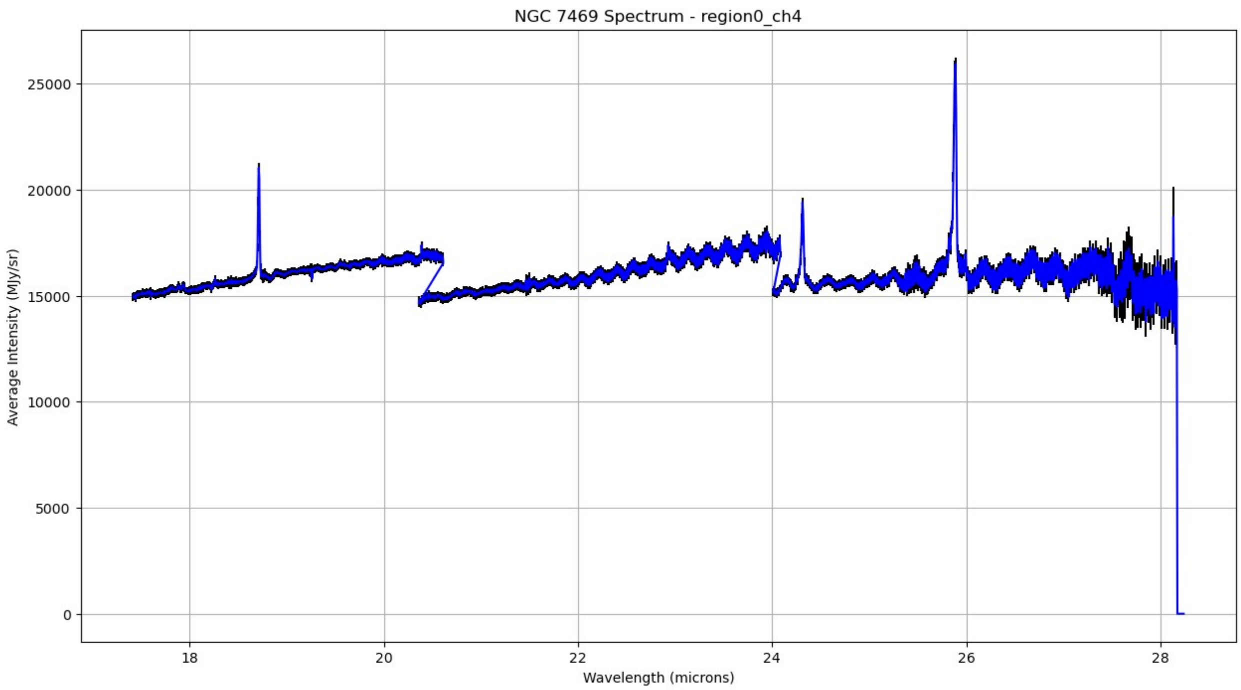

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:43.551' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.308' from MJD-AVG.
Set DATE-END to '2022-07-04T04:01:01.688' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559130 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445726.821 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:30.910' from MJD-BEG.
```

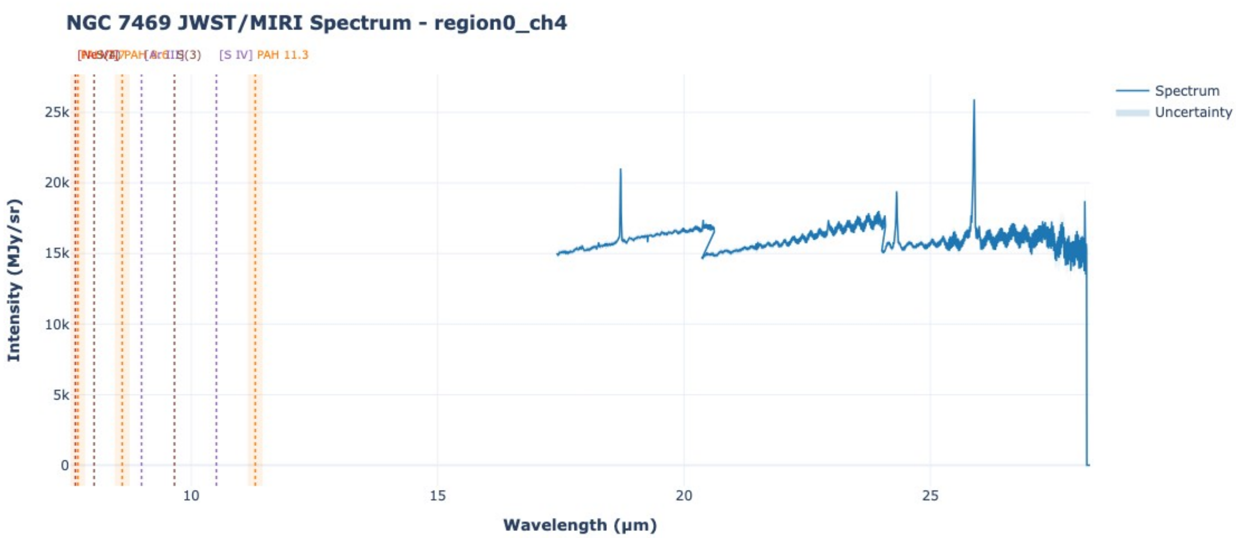
```
Set DATE-AVG to '2022-07-04T04:11:30.971' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:32.407' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557469 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283458 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461174.508 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:23.837' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.114' from MJD-AVG.
Set DATE-END to '2022-07-04T04:34:17.014' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555798 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476710.042 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```





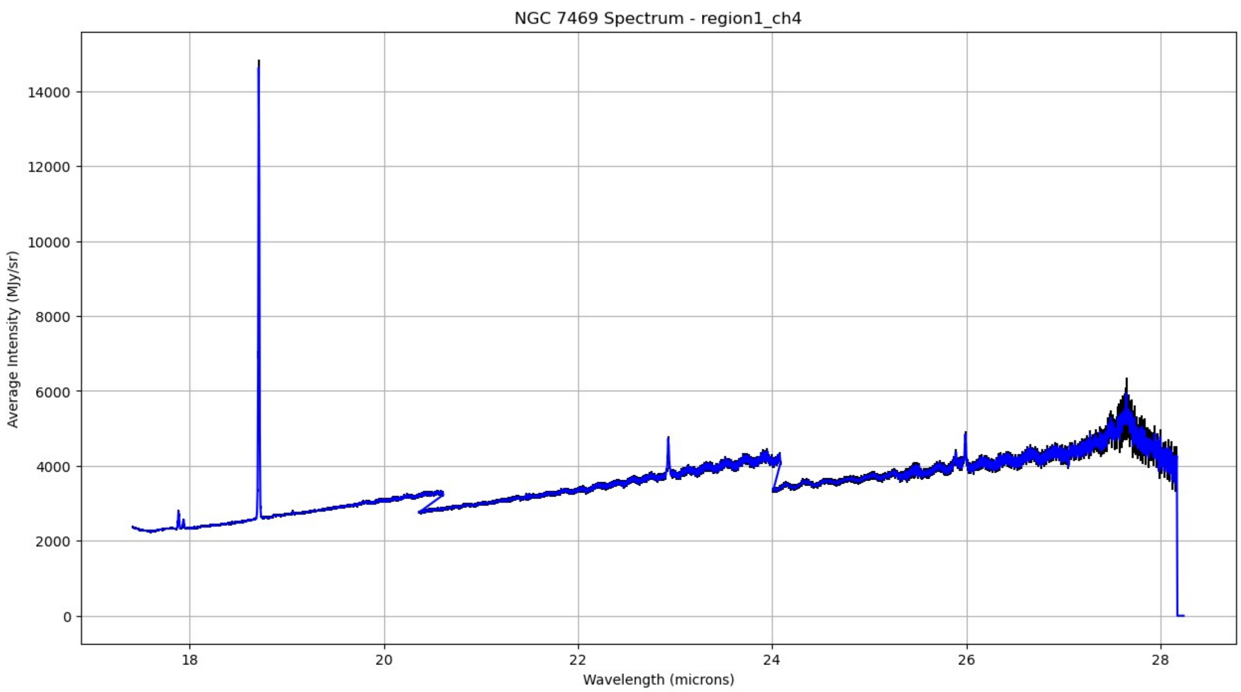

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:43.551' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.308' from MJD-AVG.
Set DATE-END to '2022-07-04T04:01:01.688' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559130 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445726.821 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:30.910' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:11:30.971' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:32.407' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557469 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283458 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461174.508 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:23.837' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.114' from MJD-AVG.
Set DATE-END to '2022-07-04T04:34:17.014' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555798 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476710.042 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```

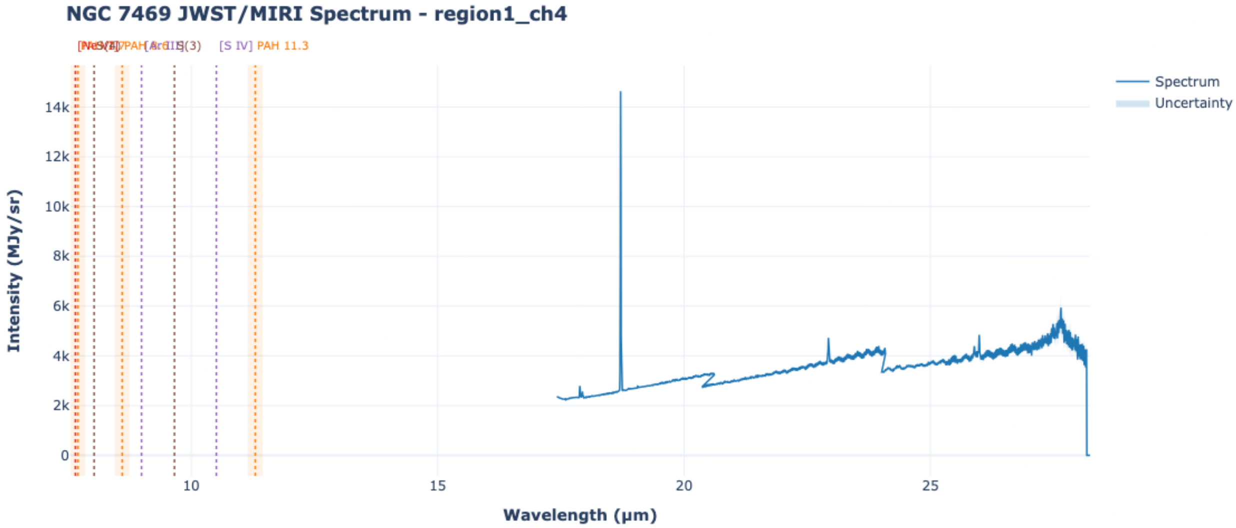





```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T03:48:43.551' from MJD-BEG.
Set DATE-AVG to '2022-07-04T03:54:53.308' from MJD-AVG.
Set DATE-END to '2022-07-04T04:01:01.688' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.559130 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.282938 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737445726.821 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:05:30.910' from MJD-BEG.
```

```
Set DATE-AVG to '2022-07-04T04:11:30.971' from MJD-AVG.
Set DATE-END to '2022-07-04T04:17:32.407' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.557469 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283458 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737461174.508 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATE-BEG to
'2022-07-04T04:22:23.837' from MJD-BEG.
Set DATE-AVG to '2022-07-04T04:28:21.114' from MJD-AVG.
Set DATE-END to '2022-07-04T04:34:17.014' from MJD-END'.
[astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'obsfix' made the change 'Set OBSGEO-L to
-72.555798 from OBSGEO-[XYZ].
Set OBSGEO-B to   -38.283980 from OBSGEO-[XYZ].
Set OBSGEO-H to 1737476710.042 from OBSGEO-[XYZ]'. [astropy.wcs.wcs]
```





Run for All Channels Combined