CALCULATING HUBBLE CONSTANT USING TYPE-1A SUPERNOVAE

Justin Lewis, Abdullahi Omar, and O'Brein Carr

MOTIVATION

- We wanted to know the speed at which the universe expands also known as the Hubble constant. In doing so we can see whether our universe is expanding at an accelerating rate or a decelerating rate. This will also help us determine the age of the universe.

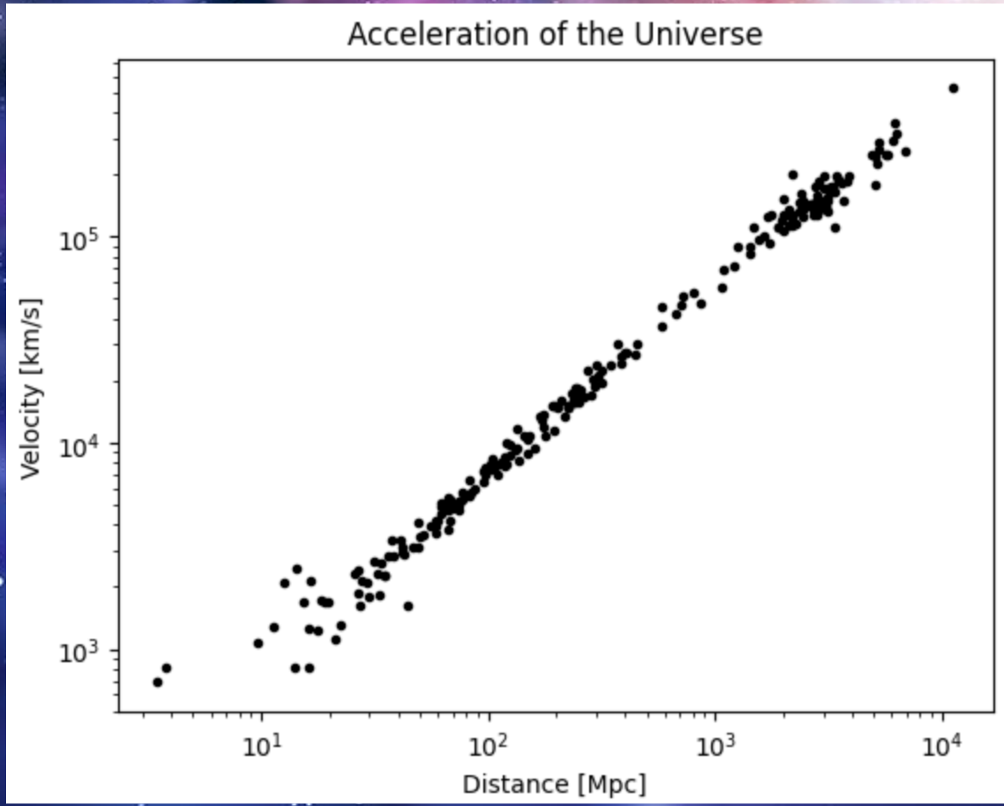
METHODS

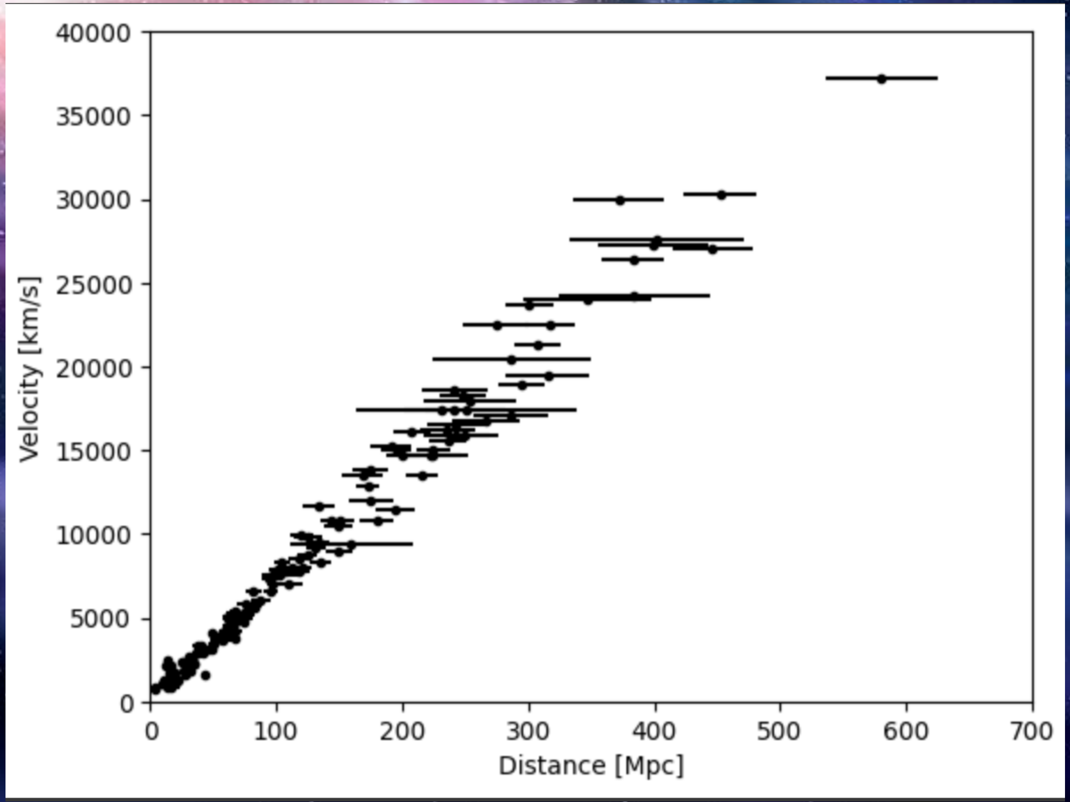
- The Slope of the linear fit is directly linked to the Hubble Constant, this relationship is found in Hubbles Law, which is used to describe the expansion of the universe.
- Hubble's Law: v = H_0 * D
- Where v is the recessional velocity, and the Law states that this velocity is proportional to its distance from the observation point(Earth).

Hubble Constant = $H_0 = \frac{v}{D}$

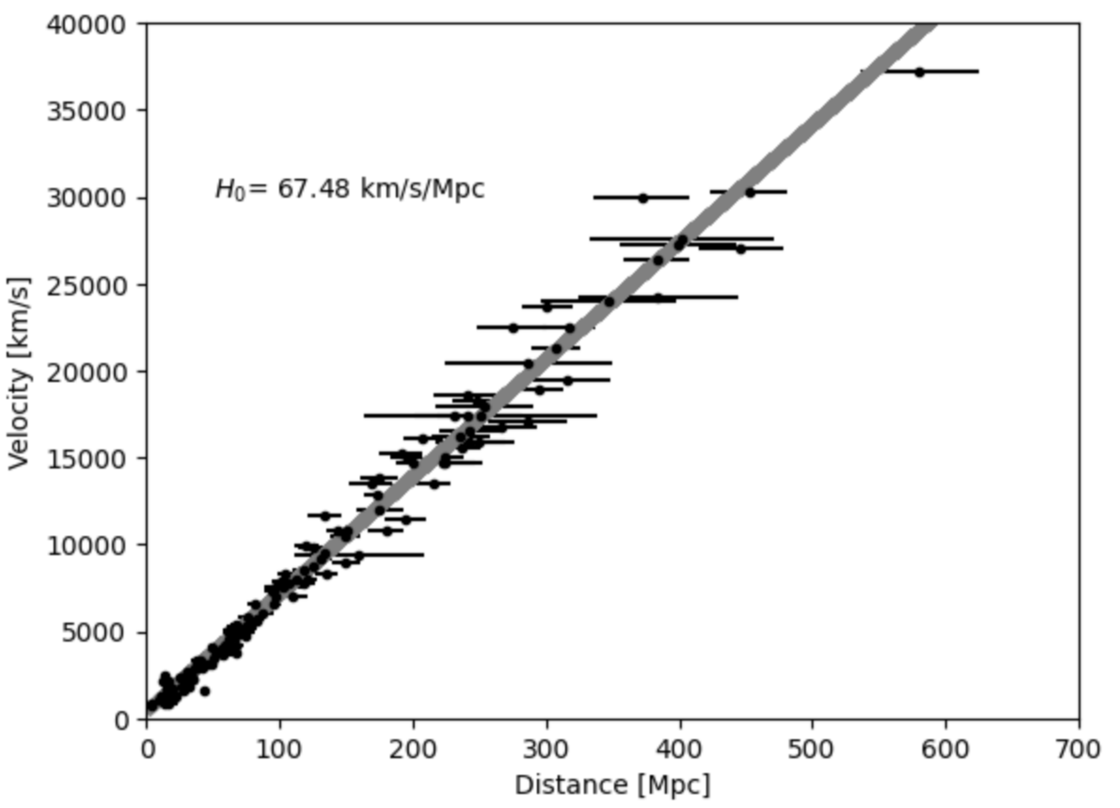
Age of the Universe is: $\frac{1}{H_0}$

RESULTS


Acceleration of the Universe
Velocity [km/s]
10^3
10^4
10^5
Distance [Mpc]
10^1
10^2
10^3
10^4


Velocity [km/s]
0
5000
10000
15000
20000
25000
30000
35000
40000
Distance [Mpc]
0
100
200
300
400
500
600
700

LINEAR GRAPH WITH HUBBLE CONSTANT



SLOPE VS AVERAGE

```
Hubble_ConstantS = velocity / distance #  This calculates the hubble constant
print(Hubble_ConstantS) # This prints the hubble constant
```

```
[11] average = np.average(Hubble_ConstantS) # This calculates the average of the hubble constant
     print(average) # This prints the average of the hubble constant in km/(mpc s)

67.7316752356907 km / (mpc s)
```

Age of universe $\approx \frac{1}{H_0}$

```
age_of_universe2 = (1 / (67.48 * (u.km/u.s/u.Mpc))).to(u.yr) # calculate the age of the universe and turns it into yr
print(age_of_universe2) # prints the age of the universe in yr

14490104055.731909 yr
```

```
14490104055.731909 - 14436262181.295479 # This is the difference between both age of the universe in years

53841874.43642998
```

This amounts to 53 million or 53841874.43642998 which is a significant amount for the years. The problem is that the slope is more accurate because it isn't as affected by the outliers as the slope is which helps give a more precise measurement.

REDUCED CHI-SQUARED

```python
observed = velocity.to(u.km / u.s).value[ind]  # Observed values

if hasattr(velocity_model, 'unit'):  # Check if velocity_model is an Astropy Quantity
    model = velocity_model[ind].to(u.km / u.s).value  # Convert to km/s if it's a Quantity
else:
    model = velocity_model[ind]  # Assume it's already in km/s if it's not a Quantity

# Define velocity error as a percentage of observed values
percentage_error = 5  # Replace with your actual percentage error estimate
velocity_error = (percentage_error / 100) * observed * u.km / u.s  # Velocity errors based on observed values

errors = velocity_error.to(u.km / u.s).value  # Convert errors to km/s if needed

reduced_chi_sq = reduced_chi_squared(observed, model, errors)  # This is the formula for the chi squared
print(f"Reduced Chi-Squared: {reduced_chi_sq}")  # Print the reduced chi-squared value
```

```
Reduced Chi-Squared: 13.52455005206
```

- Observed: Data from given data set
- Model: Values found in the linear model
- Error: Errors or uncertaintities of the observed data

The function reduced_chi_squared is the normalized version of the chi-squared statistic which includes the degrees of freedom within the given data.

CONCLUSIONS

- The slope is more accurate in obtaining the Hubble constant than the average of the Hubble constants. This is because the slope isn't affected by the numbers outside of the linear graph as much as the average.
- We also determined the Hubble constant to be 67.48 km/(Mpc s) and determined the age of the universe was approximately 14.49 billion years old.
- Finally, we determined that the reduced chi-squared was 13.52455005206.

REFERENCES/CITATIONS

- "Reduced Chi-Squared" prompt. Gemini, Google, 8 Nov. 2023, colab.research.google.com.

- Gemini was used to help create the functions within our code