Supernovas Ia y la expansión acelerada del universo

Iván Villegas Pérez

Un trabajo presentado para la asignatura *Astrofísica (G69)*

Departamento de Física Moderna
Universidad de Cantabria
España

	Einstein-de Sitter	Concordancia ΛCDM
Ω_Λ	0	0.689
Ω_m	1	0.311
Ω_r	0	0.017
Ω_k	0	-0.017

TABLE I: Valores de Ω_Λ, Ω_m, Ω_r and Ω_k en cada modelo.

I. INTRODUCCIÓN

En 2011 el Premio Nobel de Física fue concedido a Saul Perlmutter, Brian P. Schmidt y Adam G. Reiss por el descubrimiento de la expansión acelerada del universo[2] mediante supernovas de tipo Ia (Sn Ia). La idea de este trabajo es reproducir estos resultados y comprobar que la expansión del universo es una expansión acelerada.
Para ello, se han obtenido los datos medidos por los ganadores del Nobel y se ha obtenido el módulo de la distancia meidante la Ecuación 1.

$$\mu = m - M_v, \tag{1}$$

donde μ es el módulo de la distancia, m es la magnitud aparente y $M_v = -19.3$ es la magnitud absoluta en el espectro óptico.

Seguidamente, se han obtenido estimaciones teóricas de este módulo de la distancia mediante dos modelos diferentes; el modelo Einstein-de Sitter y la concordancia ΛCDM. Para ello, se ha utilizado la Ecuación 2.

$$\mu = 5 \cdot \log_{10}(d) - 5, \tag{2}$$

donde d es la distancia entre la supernova y la Tierra. El principal problema es la medida de la distancia, pues hay varios métodos, dando cada uno un resultados distintio. Para estas mediciones se ha optado por la medición de la distancia de luminosidad (D_L), que viene dada por la Ecuación 3.

$$D_L = c\frac{1+z}{H_0}\int_0^z \frac{1}{E(z')}dz', \tag{3}$$

donde c es la velocidad de la luz, z es el corrimiento al rojo, H_0 es la constante de Hubble en la actualidad ($H_0 = 67.59$ km/s Mpc) y donde $E(z)$ es la siguiente expresión:

$$E(z) = \sqrt{\Omega_m(1+z)^3 + \Omega_k(1+z)^2 + \Omega_r(1+z)^4 + \Omega_\Lambda}.$$

Para cada modelo, se tienen distintos valores de cada Ω, los cuales se pueden observar en la Tabla I. Tanto los valores de H_0 como de Ω_Λ y Ω_m han sido obtenidos del *Data Release de la colaboración Planck de 2015* (PR2-2015)[4]
El motivo por el que se ha optado por la distacia de lumnosidad y el uso de Sn Ia es por el hecho de que estas supernovas tinen todas la misma lumnisidad absoluta (que no aparente) al tener todas el mismo origen; la explosión de una enana blanca de C-O en presencia de una compañera gigante, por lo que son una buena candela estándar. Sabiendo entonces su corrimiento al rojo y el tipo de universo en el que estamos, se puede obtener la distancia (como en nuestro caso) o utilizar la distancia y el corrimiento al rojo para saber cómo es nuestro universo. La principal desventaja de este método para medir distancias es su alta sensibilidad a lso valores de H_0 y Ω_Λ, motivo por el cuál se utilizarán datos reales (procedentes del PR2-2015) y no se probará con valores asignados aleatoriamente hasta encontrar el mejor ajuste a los puntos observados.

II. PROCEDIMIENTO

En primer lugar se han obtenido los datos de los grupos de Perlmutter, Schmidt y Reiss, almacenandolos en dos ficheros distintos “datos1.dat”, pertenecientes a las observaciones SCP $\mathrm{SN_E}$ $\mathrm{I_A}$ [1] y “datos2.dat” pertenecientes a las observaciones CALÁN/TOLOLO $\mathrm{SN_E}$ $\mathrm{I_A}$.[1] Estos ficheros pueden verse en la carpeta "Data" en mi repositorio público de GitHub, el enlace puede verse en la Sección V.

Posteriormente, se han obtenido los valores de H_0, Ω_m y Ω_Λ del PR2-2015. Al dar seis valores distintos, se ha optado por hacer la media de estos para cada variable, siendo finalmente el valor utilizado $H_0 = 67.59$ km/s Mpc. Los valores de Ω_m y Ω_Λ se han utilizado para el modelo de la concordancia ΛCDM, pues en el modelo de Einstein-de Sitter, por definición, $\Omega_m \equiv 1$, mientras que $\Omega_\Lambda = \Omega_r = \Omega_k = 0$.

Para obtener la distancia de luminosidad (Ecuación 3) en cada modelo se ha necesitado realizar una integral, la cual ha sido aproximada numéricamente mediante la función 'integrate(función, varibale)' del módulo 'scipy' de Python, que está integrada en el programa disponible en el Anexo (Sección V). Una vez realizada esta primera parte, se han añadido más datos de supernovas procedentes del "Supernova Cosmology Project" (SCP)[5], que se encuentran en la carpeta "Data" de mi repositorio público de GitHub bajo el nombre "datos.dat" o en el siguiente enlace[1].

III. RESULTADOS

Una vez obtenidos los datos de "datos1.dat" y "datos2.dat" se ha realizado la Figura 1, mostrando los pares de datos obtenidos por Perlmutter y su grupo mediante la Ecuación 1. A continuación, se ha obtenido los valores teóricos del módulo de la distancia mediante la Ecuación 2, pudiendo verse los modelos Einstein-de Sitter y el modelo de concordancia ΛCDM en la Figura 2. Como se puede ver, todo parece indicar que el modelo de concordancia ΛCDM tiene mayor similitud con los puntos observados. Para salir de dudas, se han añadido más datos (Figura 3) y otro modelo (Figura 4), confirmandonos el resultado.

Se ha optado por introducir otro modelo con $\Omega_m = \Omega_\Lambda = 0.5$ una vez se ha visto que el término Ω_Λ tiene un peso mucho mayor que Ω_r u Ω_k. Aunque también tiene mayor peso que Ω_m, la diferencia entre este modelo y el de concordancia ΛCDM no es tanta. También se ha realizado la comprobación con otro modelo con $\Omega_\Lambda = 0.7$ y $\Omega_m = 0.3$, pero no resultaba diferenciable del modelo de concordancia ΛCDM.

Finalmente, se ha intentado hacer un ajuste minimizando la distancia entre posibles ajustes y los puntos experimentales. Para realizar esta operación, se ha utilizado la función "Minimiza" del módulo "MyN_python_siempre_contigo.py" [3]. El resultado puede verse en la Figura 5. Una vez más, se observa que el mejor ajuste coincide con el modelo de concordancia ΛCDM, no obstante, se obtienen los valores de $\Omega_\Lambda \approx -6.82$ y $\Omega_m \approx 7.82$, lo que indica que aunque gráficamente el ajuste es bueno, no nos da más información sobre el valor exacto de estos parámetros.

IV. CONCLUSIONES

Una vez representados los datos y comparados con distintos modelos, es trivial llegar a la conclusión de que el modelo que mejor se ajusta a los datos observacionales es el modelo de concordancia ΛCDM, lo que implicaría que el universo se encuentra en expansión acelerada debido al mayor valor de Ω_Λ, lo que implica que nos encontramos en una época de dominio del vacío y por tanto, en expansión acelerada.

V. ANEXO

El código utilizadopuede encontrarse a continuación, y tambien en mi repositorio público de GitHub junto a todos los archivos creados. En el siguiente enlace: https://github.com/ivanvillegas7/Supernovas-and-the-accelerated-espansion-of-the-universe

[1] https://supernova.lbl.gov/Union/figures/SCPUnion2.1_mu_vs_z.txt

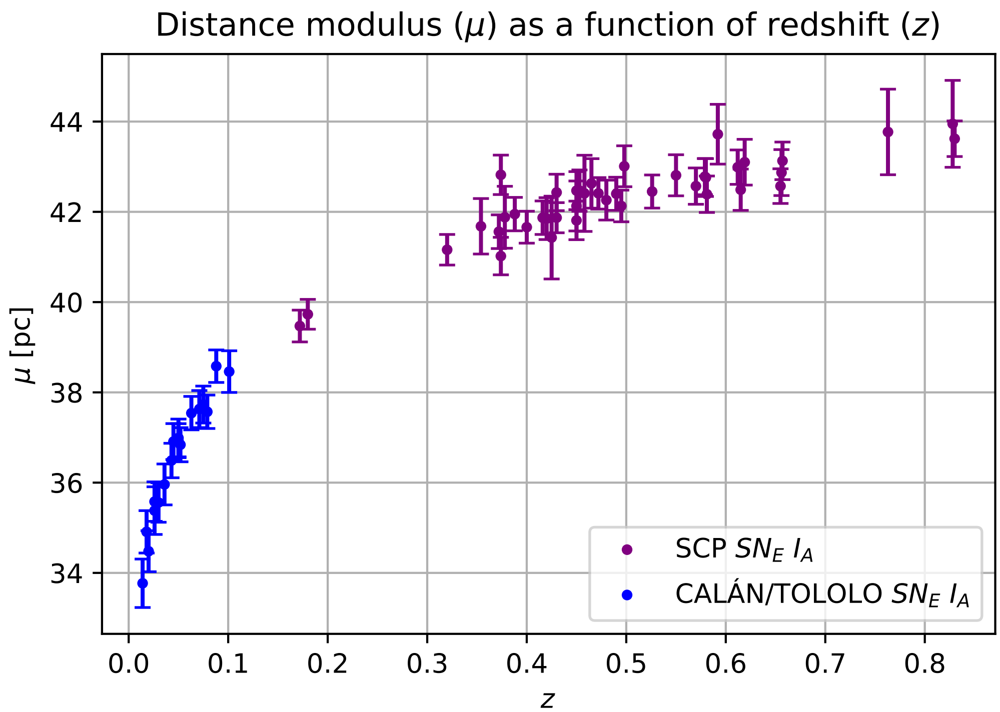

FIG. 1: Representación del módulo de la distancia (μ) frente al corrimiento al rojo (z) de las observaciones de Perlmutter.

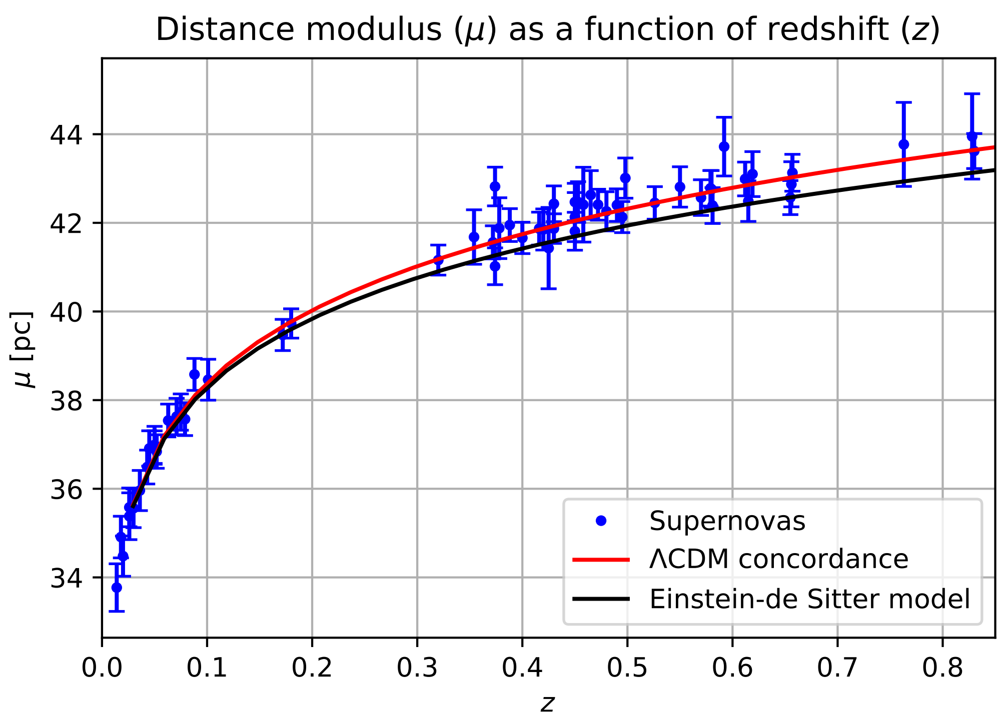

FIG. 2: Representación del módulo de la distancia (μ) frente al corrimiento al rojo (z) de las observaciones de Perlmutter. Superpuesto, pueden verse los modelos de Einstein-de Sitter y de concordancia ΛCDM

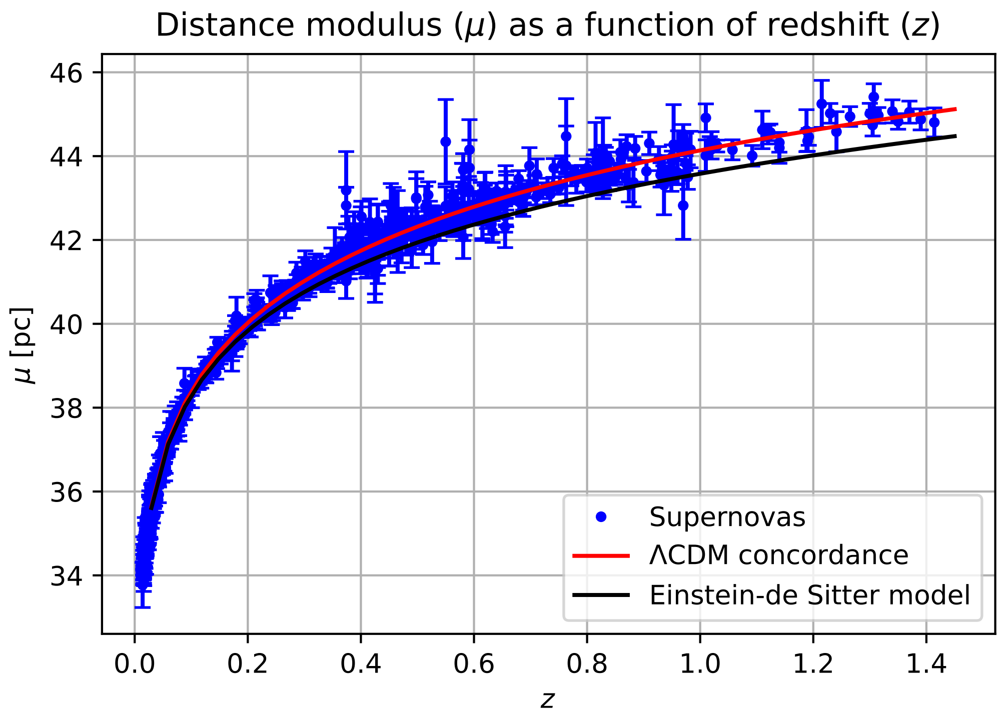

FIG. 3: Representación del módulo de la distancia (μ) frente al corrimiento al rojo (z) de las observaciones de Perlmutter y las observaciones del SCP. Superpuesto, pueden verse los modelos de Einstein-de Sitter y de concordancia ΛCDM

```
# -*- coding: utf-8 -*-
"""
Created on Sun Nov 20 15:04:05 2022

@author: Iv n Villegas P rez
"""
#%%Importamos los m dulos que se van a utilizar

from typing import List

import numpy as np

import matplotlib.pyplot as plt

import urllib.request

import pandas as pd

import scipy.integrate as integrate

from MyN_python_siempre_contigo import Minimiza

#%%Importamos los datos

M_v: float = -19.3

j: int = 0

with  open("datos1.dat", "r") as infile:

    lines = infile.readlines()
```

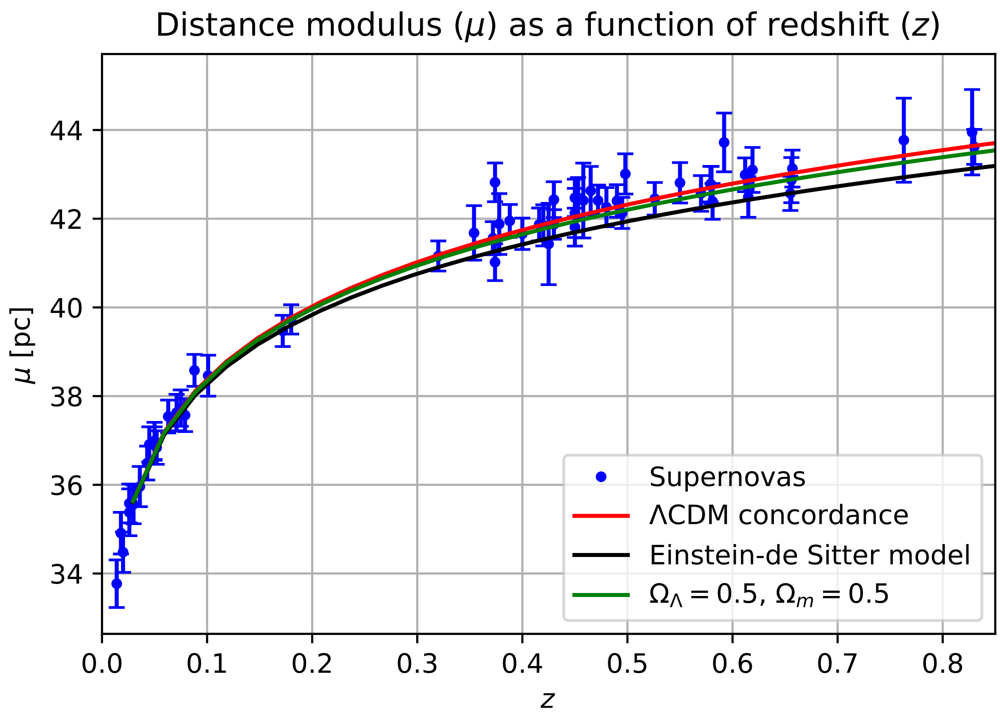

FIG. 4: Representación del módulo de la distancia (μ) frente al corrimiento al rojo (z) de las observaciones de Perlmutter. Superpuesto, pueden verse los modelos de Einstein-de Sitter, de concordancia ΛCDM y un tercero con $\Omega_\Lambda = \Omega_m = 0.5$.

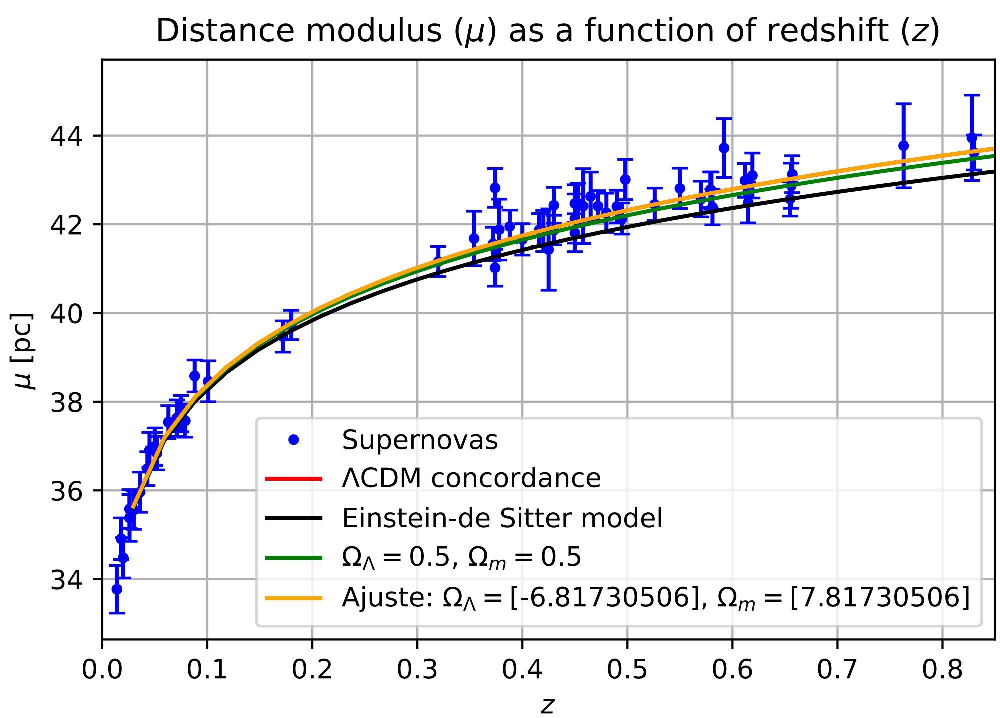

FIG. 5: Mejor ajuste para los datos experimentales, con $\Omega_\Lambda \approx -6.82$ y $\Omega_m \approx 7.82$.

```
    z1: List[str] = []
    r1: List[float] = []
    r_err1: List[float] = []
    for  line in  lines:
        #Nos saltamos la primera fila del archivo
        if j==0:
            j+=1
        else:
            vals = line.split()
            z1.append(float(vals[1]))
            r1.append(float(vals[8])-M_v)
            r_err1.append((float(vals[8])-M_v)*float(vals[9])/float(vals[8]))
k: int = 0
with  open("datos2.dat", "r") as infile:
    lines = infile.readlines()
    z2: List[str] = []
    r2: List[float] = []
    r_err2: List[float] = []
    for  line in  lines:
        #Nos saltamos la primera fila del archivo
        if k==0:
            k+=1
        else:
            vals = line.split()
            z2.append(float(vals[1]))
            r2.append(float(vals[8])-M_v)
            r_err2.append((float(vals[8])-M_v)*float(vals[9])/float(vals[8]))
targetURL = "https://supernova.lbl.gov/Union/figures/SCPUnion2.1_mu_vs_z.txt"
i = 0
z = []
r = []
r_er = []
for line in urllib.request.urlopen(targetURL):
    #Nos saltamos las 5 primeras filas del archivo
    if i <= 5:
```

```
        column_names = line

        i+=1

    else:

        z.append(float(line.split()[1]))

        r.append(float(line.split()[2]))

        r_er.append(float(line.split()[3]))

output_df = pd.DataFrame({column_names[:17]: z, column_names[27:-1]: r,\
                          column_names[27:-1]: r_er})

output_df.to_csv('datos.dat')

#%%Tarea1

plt.figure()


plt.plot(z1, r1, '.', label=r'SCP $SN_E$ $I_A$', color='purple')

for i in range(len(r1)):

    plt.errorbar(z1[i],r1[i], yerr = r_err1[i], capsize = 3, color='purple')

plt.plot(z2, r2, '.', label=r'CAL N/TOLOLO $SN_E$ $I_A$', color='blue')

for i in range(len(r2)):

    plt.errorbar(z2[i],r2[i], yerr = r_err2[i], capsize = 3, color='blue')

plt.xlabel(r'$z$')

plt.ylabel(r'$\mu$ [pc]')

plt.title(r'Distance modulus ($\mu$) as a function of redshift ($z$)')

plt.legend()

plt.grid(True)


plt.savefig('r frente a z - sin modelos.pdf')

#%%Tarea2

H0: float = (67.31+67.81+67.9+67.27+67.51+67.74)*1e-3/6 # km/s Mpc -> m/s pc

c: float = 3e8 #m/s

Vacio: float = (0.685+0.692+0.6935+0.6844+0.6879+0.6911)/6

Materia: float = (0.315+0.308+0.3065+0.3156+0.3121+0.3089)/6

Curvatura: float = (0.0008-0.052-0.005-0.0001-0.04-0.004)/6

Radiacion: float = 1-Vacio-Materia-Curvatura

CDM: List[float] = []

def E_CDM(x: np.array)-> np.array:

    return 1/np.sqrt(Materia*(1+x)**3+Vacio+Radiacion*(1+x)**4+Curvatura*(1+x)**2)

EdS: List[float] = []

def E_EdS(x: np.array)-> np.array:
```

```
    return 1/np.sqrt((1+x)**3)
C_C: List[float] = []
def E_C_C(x: np.array)-> np.array:
    return 1/np.sqrt(0.5*(1+x)**3+0.5)
z_o: np.array = np.linspace(0, 1.45)
for i in range(len(z_o)):
    x: np.array = np.array(np.linspace(0, z_o[i]))
    CDM.append(5*np.log10(((c/H0)*(1+z_o[i])*integrate.simpson(E_CDM(x), x)))-5)
    EdS.append(5*np.log10(((c/H0)*(1+z_o[i])*integrate.simpson(E_EdS(x), x)))-5)
    C_C.append(5*np.log10(((c/H0)*(1+z_o[i])*integrate.simpson(E_C_C(x), x)))-5)

plt.figure()

plt.plot(z1, r1, '.', label='Supernovas', color='blue')
for i in range(len(r1)):
    plt.errorbar(z1[i],r1[i], yerr = r_err1[i], capsize = 3, color='blue')
plt.plot(z2, r2, '.', color='blue')
for i in range(len(r2)):
    plt.errorbar(z2[i],r2[i], yerr = r_err2[i], capsize = 3, color='blue')
plt.plot(z_o, CDM, color='red', label='$\Lambda$CDM concordance')
plt.plot(z_o, EdS, color='black', label=r'Einstein-de Sitter model')
plt.xlabel(r'$z$')
plt.ylabel(r'$\mu$ [pc]')
plt.title(r'Distance modulus ($\mu$) as a function of redshift ($z$)')
plt.legend()
plt.xlim(0, 0.85)
plt.grid(True)

plt.savefig('r frente a z.pdf')
#%%Objetivos avanzados 1
plt.figure()

plt.plot(z1, r1, '.', label='Supernovas', color='blue')
for i in range(len(r1)):
    plt.errorbar(z1[i],r1[i], yerr = r_err1[i], capsize = 3, color='blue')
plt.plot(z2, r2, '.', color='blue')
for i in range(len(r2)):
```

```
    plt.errorbar(z2[i],r2[i], yerr = r_err2[i], capsize = 3, color='blue')
plt.plot(z, r, '.', color='blue')
for i in range(len(r)):
    plt.errorbar(z[i],r[i], yerr = r_er[i], capsize = 3, color='blue')
plt.plot(z_o, CDM, color='red', label='$\Lambda$CDM concordance')
plt.plot(z_o, EdS, color='black', label=r'Einstein-de Sitter model')
plt.xlabel(r'$z$')
plt.ylabel(r'$\mu$ [pc]')
plt.title(r'Distance modulus ($\mu$) as a function of redshift ($z$)')
plt.legend()
plt.grid(True)

plt.savefig('r frente a z extra datos.pdf')
#%%Objetivos avanzados 2
plt.figure()

plt.plot(z1, r1, '.', label='Supernovas', color='blue')
for i in range(len(r1)):
    plt.errorbar(z1[i],r1[i], yerr = r_err1[i], capsize = 3, color='blue')
plt.plot(z2, r2, '.', color='blue')
for i in range(len(r2)):
    plt.errorbar(z2[i],r2[i], yerr = r_err2[i], capsize = 3, color='blue')
plt.plot(z_o, CDM, color='red', label='$\Lambda$CDM concordance')
plt.plot(z_o, EdS, color='black', label=r'Einstein-de Sitter model')
plt.plot(z_o, C_C, color='green', label='$\Omega_\Lambda=0.5$, $\Omega_m=0.5$')
plt.xlabel(r'$z$')
plt.ylabel(r'$\mu$ [pc]')
plt.title(r'Distance modulus ($\mu$) as a function of redshift ($z$)')
plt.legend()
plt.xlim(0, 0.85)
plt.grid(True)

plt.savefig('r frente a z extra modelos.pdf')
#%%Objetivos avanzados 3
# Buscamos valores   _m  y   _
z_exp = z1 + z2
```

```
r_exp = r1 + r2

def E_aj(x: np.array, o_m: float, o_v: float):

    return 1/np.sqrt(o_m*(1+x)**3+o_v)

def f(o_m):

    o_v = 1-o_m

    r_aj = 5*np.log10((c/H0)*(1+z_o)*integrate.simpson(E_aj(z_o, o_m, o_v), z_o))-5

    diferencia = 0

    for iexp in range(len(z_exp)):

        dif = np.inf

        for iaj in range(len(z_o)):

            difx = z_exp[iexp]-z_o[iaj]

            dify = r_exp[iexp]-r_aj[iaj]

            d = difx**2+dify**2 # Sin ra z para minimizar errores de c lculo

            if d<dif:

                dif = d

        diferencia+=dif

    return diferencia

Materia_aj = 0

h = 1e-3

epsilon = 1e-8

N = 200

Materia_aj, f_Materia_aj = Minimiza(f, np.array([Materia_aj]), h, epsilon, N)

Vacio_aj = 1 - Materia_aj

omega_k_ajuste = 0

omega_r_ajuste = 0

r_ajuste1 = 5*np.log10((c/H0)*(1+z_o)*integrate.simpson(E_aj(z_o, Materia_aj, Vacio_aj), z_o))-
    5

plt.figure()

plt.plot(z_exp, r_exp, '.', label='Supernovas', color='blue')

for i in range(len(r1)):

    plt.errorbar(z1[i],r1[i], yerr = r_err1[i], capsize = 3, color='blue')

plt.plot(z2, r2, '.', color='blue')

for i in range(len(r2)):

    plt.errorbar(z2[i],r2[i], yerr = r_err2[i], capsize = 3, color='blue')

plt.plot(z_o, CDM, color='red', label='$\Lambda$CDM concordance')
```

```
plt.plot(z_o, EdS, color='black', label=r'Einstein-de Sitter model')

plt.plot(z_o, C_C, color='green', label='$\Omega_\Lambda=0.5$, $\Omega_m=0.5$')

plt.plot(z_o, CDM, color='orange', label=f'Ajuste: $\Omega_\Lambda=${Vacio_aj}, $\Omega_m=${
                                                  Materia_aj}')

plt.xlabel(r'$z$')

plt.ylabel(r'$\mu$ [pc]')

plt.title(r'Distance modulus ($\mu$) as a function of redshift ($z$)')

plt.legend()

plt.xlim(0, 0.85)

plt.grid(True)


plt.savefig('r frente a z ajuste.pdf')
```

BIBLIOGRAPHY

[1] S. Perlmutter et al. "Measurements of and from 42 High-Redshift Supernovae". In: *The Astrophysical Journal* 517.2 (June 1999), p. 565. DOI: 10.1086/307221. URL: https://dx.doi.org/10.1086/307221.

[2] Royal Swedish Academy of Sciences. "THE ACCELERATING UNIVERSE". In: (2011). URL: https://www.nobelprize.org/uploads/2018/06/advanced-physicsprize2011.pdf.

[3] Carlos Beltán Álvarez. *MyN_python_siempre_contigo.py*. 2021.

[4] European Space Agency. *Planck Legacy Archive*. URL: http://pla.esac.esa.int/pla/#cosmology. (acceso: 20.11.2022).

[5] Supernova Cosmology Project. *Introduction to the SCP Union2.1 Compilation*. URL: https://supernova.lbl.gov/Union/. (acceso: 20.11.2022).