LEY DE HUBBLE

Proyecto de fin de curso IFC

María Salazar
Adrián Sanjurjo

Contenido

Resumen de contenido

Puesto que la física es una ciencia observacional, en gran parte de ella entra en juego la medición de datos e incertidumbres, y mediante ello poder sacar unas conclusiones. En esta parte entra en juego la estadística, que nos permite jugar con los datos en la obtención de los resultados, sacando valores estimados y poder concluir leyes físicas. Estas observaciones llegan a ámbitos tales como la Astrofísica.

1. Tras el desarrollo de la teoría de la Relatividad General de Einstein, muchos científicos dieron un salto y utilizaron esta rama para el estudio del comportamiento de los objetos celestes a grandes distancias. Uno de ellos fue Erwing Hubble, cuyos estudios y observación del cosmos lograron resultados en la expansión del Universo. Uno de estos efectos es que las galaxias y cúmulos estelares, a cuanta mayor distancia se encuentren entre ellos, mayores velocidades de alejamiento tendrán. A pesar de los pocos recursos de observación cosmológica de la época, logró hallar una relación entre estos dos parámetros, velocidad respecto a la distancia, mediante una constante denominada en su honor la Constante de Hubble. A medida que han mejorado estas técnicas, se ha logrado dar un valor cada vez más preciso de dicha constante, teniendo a lo largo de la historia un valor entre 70 y 500 $Km \cdot Mpa \cdot s^{-1}$.

En nuestra práctica en Python, mostraremos un análisis de mediciones y resultados para hallar el valor más adecuado de esta constante, a partir de datos del propio Hubble y otros más recientes de la década de los 90.

Introducción y desarrollo

Este estudio nos ayuda a entender la creación de espacio entre galaxias, y comprender la expansión del universo postulada por George Lemâitre, cuyos estudios rivalizaron con el universo estático predicho por Einstein y su constante cosmológica. Los resultados observacionales fueron en contra del padre de la Relatividad, quien admitió que la constante cosmológica fue su mayor error. Esto fue visto mediante el *"Redshift"* de la luz, su efecto Doppler, el cual hace que las ondas electromagnéticas disminuyan su frecuencia (aumenta la longitud de onda) a medida que se alejan del sistema de referencia desde el cual se observan, produciendo un efecto llamado *"Corrimiento al rojo"*, mientras que si se acercasen, aumentaría su frecuencia, como viene en la relación de la energía de Plank:

$$E_\gamma = h \cdot \nu = \frac{h \cdot c}{\lambda}$$

Siendo ν la frecuencia de la onda, *c* la velocidad de la luz en el vacío y λ la longitud de onda.

El efecto Redshift se describe mediante la fórmula:

$$z = \frac{\lambda_r - \lambda_e}{\lambda_e}$$

Con λ_r la longitud de onda recibida y λ_e la longitud emitida por el cuerpo celeste.

Todo esto viene ligado a las velocidades de alejamiento de dichos cuerpos. Nuestro objetivo es conseguir comprobar la validez de la relación de Hubble entre distancia y velocidad, basándonos en datos observacionales y métodos estadísticos para la determinación de la constante que las relaciona. En este caso se trata de un ajuste lineal, por lo que no es excesivamente complicado.

Para ello, usaremos dos funciones de Python, que nos aproximan por mínimos cuadrados la pendiente (la constante) y así calcular los valores esperados. Con datos medidos por Hubble, y continuadamente, con datos más recientes tomados por el telescopio espacial Hubble, llamado así en su honor, compararemos la aproximación del valor que deberá tener, así como la bondad de dichos ajustes. El ajuste por mínimos se realiza mediante las funciones Least_Square y Curve_Fit, siendo su fórmula:

$$m = \frac{n \cdot (\sum_{i=1}^{n} x_i \cdot y_i) - (\sum_{i=1}^{n} x_i) \cdot (\sum_{i=1}^{n} y_i)}{n \cdot (\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2}$$

Contamos pues para este cálculo estas funciones matemáticas:

1. Ley de Hubble:

$$V = H_0 \cdot D$$

V la velocidad en km· s^{-1}; H_0 la constante de Hubble; *D*la distancia en MPc

2. Desviación típica (incluyendo el cálculo de la media y las desviaciones):

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N}}$$

Siendo $x_i - \bar{x}$ la desviación respecto a la media; N número de valores.

3. Bondad del ajuste:

$$x^2 = \sum_{i=1}^{N} \frac{(x_{i-obs} - x_{i-esp})^2}{(x_{i-obs} - \bar{x})^2}$$

Descripción del programa

El programa consiste principalmente de cinco partes:

1. Definición de funciones.
2. Recogida de datos de Hubble y cálculo del ajuste.
3. Recogida de datos de los años 90 y repetición del cálculo.
4. Muestreo por pantalla mediante gráficas (uso de matplotlib) y comparación de resultados.
5. Representación en Visual de dos galaxias alejándose, presentando el efecto del corrimiento al rojo

Comentaremos detalladamente cada uno de estos pasos, en los que usaremos los módulos Numpy, Matplotlib, Scipy, Visual y Astropy.

- Definición de funciones

  Comenzamos con declaración de las funciones estadísticas a usar, mencionadas ya anteriormente. En la función de la Ley de Hubble, tomamos la constante como parámetro de entrada, puesto que queremos ver como varían los resultados dependiendo que valor usar. En ninguna de dichas funciones se usarán variables globales.

- Primera recogida de datos

  Explotando el módulo Numpy, recogemos los datos de Hubble desde un archivo .txt, en el cual hay tres columnas: La primera de ellas corresponde al nombre de las galaxias observadas, en formato número; la segunda a las distancias a la tierra; la tercera a sus velocidades. Estos datos se manejan en arrays, facilitando su entrada en las funciones. Vamos cambiando la forma de los arrays para que cuadren con las funciones Least_Square y Curve_Fit. Calculamos así el valor de la constante de Hubble. Una vez obtenido los valores, calculamos las velocidades estimadas a

través de dichos números, unas velocidades mediante la primera y otras por la segunda. Tras ello, medimos la exactitud, el cual es dado por la función Curve_Fit, pero el Least_Square lo hallamos mediante la función de bondad de ajuste, anteriormente expresada. Una vez obtenido todo esto, empleamos el módulo Astropy, puesto que tiene una función llamada Table, que nos permite la creación de tablas de datos fácilmente. Introducimos en ella los datos (velocidades, distancias, desviaciones respecto a la media y estimaciones de la velocidad por ambos métodos) y los muestra por pantalla. Por último, imprimimos los valores de las constantes según un método u otro.

- Segunda recogida de datos

Realizamos un proceso idéntico al del anterior apartado, a excepción de la lectura, que se realiza mediante un archivo .csv. Estos datos son más recientes y se toman distancias mucho mayores, del orden de 10 Mpc, mientras que el valor máximo de la primera toma de datos es del orden de 2 Mpc como máximo.

Mismo muestreo, no hay cambio en el procedimiento.

- Representación gráfica

En esta parte usamos el módulo Matplotlib. Para ello creamos un subplot de dos gráficas, una encima de la otra: La primera corresponde a los datos antiguos y la segunda a los más recientes. En ambas se representan los datos dispersos de las observaciones, velocidad de alejamiento frente a la distancia. También se muestran los valores de los ajustes por Least_Square y Curve_Fit, como líneas de ajuste, para así poder comprobar el resultado. En honor a Hubble, hemos resaltado los valores de velocidad que presentaría la galaxia Andrómeda por ambos métodos, aunque como bien sabemos, dicha galaxia está acercándose a la nuestra, lo que provocará una colisión.

- Representación Visual

Vamos a representar el movimiento de dos galaxias en el espacio. Para ello vamos a utilizar el módulo Visual Python.

En primer lugar, definimos tres objetos: una esfera y dos elipsoides. La esfera será la tierra, a la que ponemos la textura de la superficie terrestre. Los dos elipsoides serán las galaxias, que vamos a texturizar de forma personalizada. Para ello, importamos ‘Image’ y con ‘Image.open’ abrimos las fotos que queremos hacer textura. Definimos las texturas y se las aplicamos a los dos elipsoides, uno a cada uno.

Posicionamos la tierra en la posición (0, 0, 0) y las galaxias a una distancia de la tierra proporcional a la distancia real que obtuvimos en la segunda recogida de datos. Después creamos un bucle indefinido con “while” en el que definimos la velocidad de cada galaxia como su posición por la constante de Hubble que calculamos anteriormente y la posición siguiente como la posición anterior más la velocidad por el tiempo. Además, añadimos una etiqueta a cada galaxia que nos dé la información sobre su posición y su velocidad.

Por último, para representar el corrimiento al rojo del que hablamos antes, introducimos en el bucle dos “if” (uno para cada galaxia) en los que si la posición de los elipsoides pasa de una determinada distancia, se vuelven de color rojo, simulando como se verían en realidad.

Aplicación

El programa permite calcular la constante de Hubble para cualquier medición de datos, ya que, como hemos podido ver, nos hace el ajuste tanto para los datos recogidos por Hubble como para los datos recogidos en 1991. Además, permite predecir el movimiento de las galaxias en el espacio, dándole las coordenadas actuales y aplicando la constante de Hubble para calcular la velocidad.

Discusión de resultados

Los resultados que hemos obtenido son bastante satisfactorios. Para los datos recogidos por Hubble, los resultados fueron: 406 ± 13 (km/s)/Mpc (least square) y 381±25 (km/s)/Mpc (curve fit). El resultado es menor que el que dio Hubble (aproximadamente 500(km/s)/Mpc), ya que al incluir los datos en el programa, despreciamos aquellos que se alejaban mucho del resto. Por otro lado, para los datos de 1991, los resultados fueron: 77 ± 33 (km/s)/Mpc (least square)y 72 ± 4(km/s)/Mpc (curve fit).

La incertidumbre en el caso de los datos antiguos es menor en el resultado obtenido con least square. Por el contrario, para los datos más actuales, la incertidumbre es mucho menor con curve fit. Esta diferencia no nos permite saber que método es mejor para realizar el ajuste, aunque podemos predecir que cada uno es bueno según los datos que se le aporten.

Bibliografía

- Final Results from the Hubble Space Telescope Key Project to Measure the Hubble Constant, W. L. Freedman, B. F. Madore, B. K. Gibson, etc. (https://arxiv.org/abs/astro-ph/0012376)

- Apuntes Introducción a la Física Computacional (Campus Virtual)
- Efecto redshift (https://universocuantico.wordpress.com/2009/01/30/efectos-doppler-y-corrimientos-al-rojo-redshift/)