Métodos Numéricos para la Ciencia e Ingeniería
Informe Tarea 09

Benjamín Guerrero
18.862.370-0
24 de Noviembre, 2015

1. Pregunta 1

1.1. Introducción

En este problema, se va a usar un archivo llamado *hubble_original.dat*, que contiene 2 columnas de datos relacionados con las galaxias lejanas a la tierra, medidos por Hubble. La primera columna muestra las distancias entre la tierra y las galaxias (medidas por el método de las Cefeidas) en Mega parsecs, y la segunda columna muestra la velocidad de recesión de esas galaxias en kilómetros por segundo. El modelo usado fue el siguiente:

$$v = H_0 * D$$

Aquí, H_0 es la constante de Hubble, y se expresa en unidades de $\frac{km}{s}\frac{1}{Mpc}$.

Se pide derivar la constante de Hubble, usando los datos del archivo, e incluir el intervalo de confianza al 95%.

1.2. Procedimiento

Para hacer lo pedido, se crea un programa de nombre *Hubble.py*.

En este, primero se definen 2 funciones que se deben minimizar:

$$v = H_0 * d$$

$$d = \frac{v}{H_0}$$

El valor H_0 va a dar distinto en las dos definiciones, así que se va a usar el promedio entre los valores que resultan de las 2 funciones.

Ahora, para fitear las funciones, se va a usar *curve_fit* (visto en clases), con una estimación inicial de 2 (También se puede usar χ^2 para realizar un fiteo manual, pero en este caso se va a preferir el uso de un código más simple).

En el main, primero se extraen los datos del archivo (usando *np.loadtxt*), y se hace *curve_fit* con los datos adecuados para cada función, para extraer H_0 (En el programa, se definieron funciones a minimizar exactamente iguales a las ya definidas. Esto se hizo como parte del procedimiento para definir *curve_fit*, y no para que el código parezca redundante).

Luego, se promedian los 2 H_0 extraídos, y el número resultante va a ser el H_0 que se quería obtener.

Ahora, para extraer el intervalo de confianza, como no se incluyeron los errores de medición, se usa una simulación *bootstrap*, también vista en clases.

Para esto, se crea una función llamada *bootstrap*, que recibe los datos extraidos y el H_0 calculado, y retorna los intervalos de confianza al 95% del valor. Esto se hace mediante iteración. Primero, se generan unos datos falsos al azar entre 0 y N (siendo N el número de mediciones hechas en el archivo), como arrays de tamaño N, usando *np.random.randint*. Luego, con esos datos falsos, se calcula H_0 para esos datos (usando *curve_fit* con las 2 funciones hechas y promediando los resultados, tal como para los datos reales), y se guarda en una lista. Esto se repite 10000 veces, para asegurar la precisión de la simulación.

Luego, se ordenan los valores de H_0 calculados de esta forma, y se definen los límites bajos y altos entre el 2,5% y el 97,5%. Luego, se retorna el intervalo entre estos límites.

Para tener un mejor entendimiento de los resultados, estos se van a graficar. Primero, se define una función llamada *graficar*, que toma los datos extraídos del archivo, y los valores de H_0 calculados (para las 2 funciones, y el promedio), y retorna un gráfico distancia v/s velocidad llamado *hubble.jpg*.

Para graficar la simulación bootstrap, antes de ordenar los valores H_0 calculados con la simulación, se genera un histograma de 30 bins con esos datos. Como referencia, se agrega el valor H_0 obtenido realmente como una línea recta. Este gráfico se guarda como *bootstrap_1.jpg*.

1.3. Resultados

Después de ejecutarse, el programa retorna que, aproximadamente:

$$H_0 = 472{,}14\ \frac{km}{s}\frac{1}{Mpc}$$

El intervalo de confianza al 95% es, aproximadamente:

$$[363{,}48 : 602{,}54]$$

Finalmente, los gráficos muestran lo siguiente:

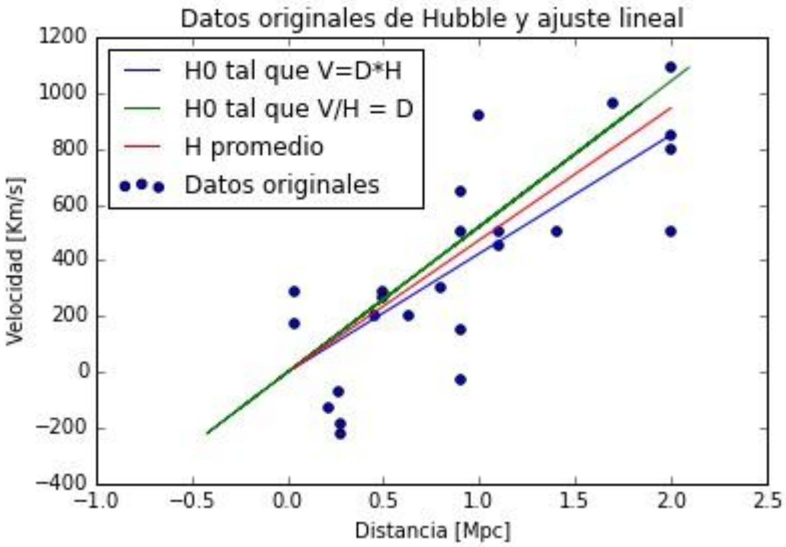

Fig 1: Comparación entre datos medidos y fiteos mediante *curve_fit*, para los datos originales de Hubble.

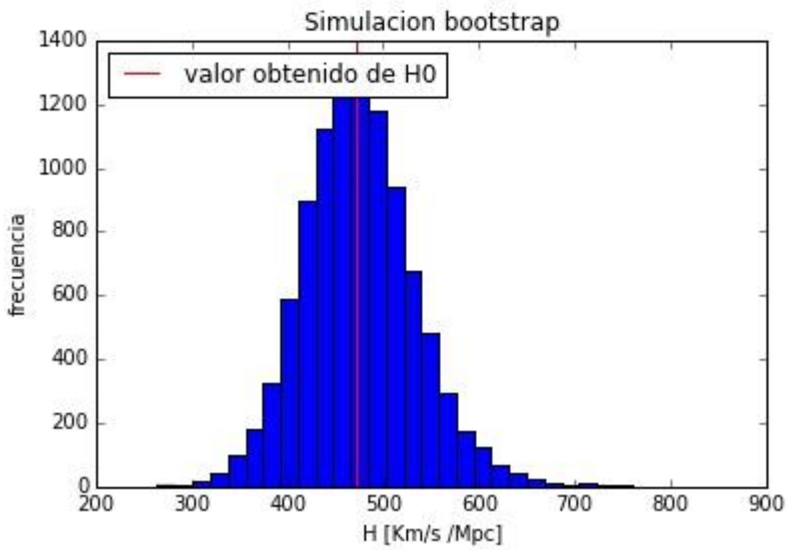

Fig 2: Histograma de la simulación Bootstrap.

2. Pregunta 2

2.1. Introducción

Nuevamente se pide calcular H_0, pero esta vez, en lugar de las mediciones obsoletas de Hubble, los datos que se van a usar son de una medición moderna y mucho más precisa, descrita en el archivo *SNIa.dat*.

Tal como en el caso anterior, también se pide calcular el intervalo de confianza al 95%.

2.2. Procedimiento

Para resolver el problema, se crea un archivo llamado *bootstrap.py*. Ahora, este archivo va a ser idéntico al anterior. Las únicas diferencias son el archivo a extraer (los datos buscados están en las columnas 2 (velocidad) y 3 (distancia)), y algunas etiquetas en los gráficos. En resumen, se va a usar *curve_fit* para fitear la función, y se hace una simulación bootstrap para calcular el intervalo de confianza. Esta vez los gráficos se llaman *SNIa.jpg*, y *bootstrap_2.jpg*.

2.3. Resultados

Una vez ejecutado, el programa muestra que, aproximadamente:

$$H_0 = 70{,}84$$

$$I_{confianza} = [67{,}96 : 74{,}56]$$

Y los gráficos generados son:

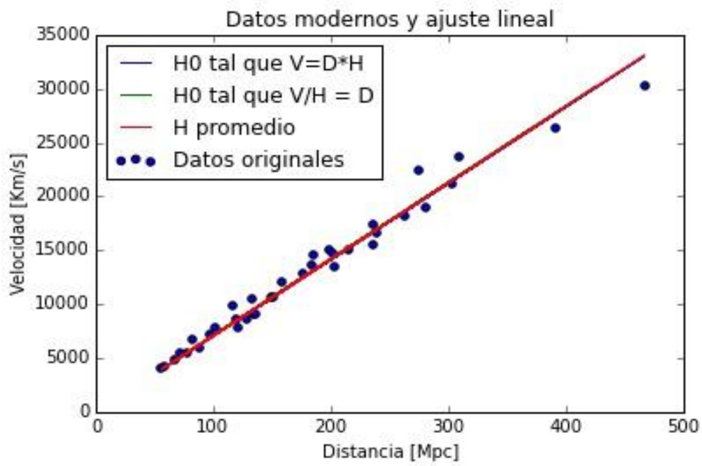

Fig 3: Comparación entre datos modernos y fiteos con *curve_fit*.

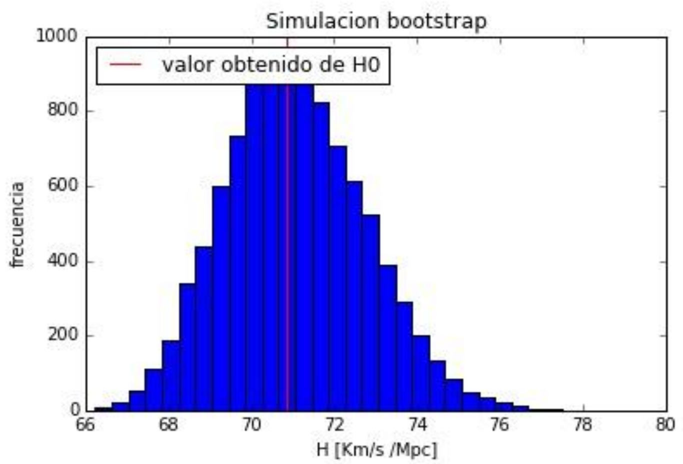

Fig 4: Simulación Bootstrap de lo anterior.

3. Pregunta 3

3.1. Introducción

Se pide encontrar la línea recta que mejor modela la relación entre el flujo en la banda i y en la banda z, para una sección recortada de un catálogo de cuásares, descrita en el archivo *DR9Q.dat*. En este archivo, el flujo en la banda i se encuentra en la columna 81 y su error asociado en la columna 82. La banda z y su error se ubican en las columnas 83 y 84, respectivamente.

También se pide el intervalo de confianza al 95%.

3.2. Procedimiento

Para este ejercicio, se crea un archivo llamado *MC.py*. El nombre del archivo se debe al método que se va a usar para extraer el intervalo de confianza.

En el main, primero se extraen los datos pedidos del archivo y se ponen en np.arrays. Luego, cada uno de estos arrays se multiplica por 3,631 para convertir las unidades de nmaggies en $\mu Jy = Jy * 10^{-6}$. Después, se va a usar la función *np.polyfit* para fitear los datos (Es un polinomio de grado 1, así que sirve). El fiteo retorna 2 números: La pendiente y la constante de la ecuación lineal.

Tal como en los casos anteriores, se va a graficar el resultado. Se usa una función *graficar* similar a la de los ejercicios anteriores. La diferencia es que aquí se va a usar *errorbar* para graficar los errores.

Ahora, como tenemos los errores asociados, se puede usar el método de Monte Carlo para generar el intervalo de confianza. Esto se hace generando un array de números aleatorios r con distribución normal entre 0 y 1, y calcular:

$$m_i = banda_i + r * error_i$$

$$m_z = banda_z + r * error_z$$

Luego, se aplica *polyfit* a los 2 resultados, y los valores de la pendiente y de la constante que se obtienen se guardan. Para garantizar la precisión, este paso se repite 10000 veces.

Ahora, se pueden graficar los resultados de manera similar a lo hecho en las partes anterior, usando histogramas. La diferencia es que esta vez, se usarán 2 gráficos en lugar de uno, porque se hace uno para la pendiente (*mc_1.jpg*) y otro para la constante (*mc_2.jpg)*.

Finalmente, se ordenan los valores de la pendiente y de la constante y se calculan los intervalos de confianza de forma similar a los problemas anteriores.

3.3. Resultados

Una vez ejecutado el programa, se tiene que la relación queda más o menos de la siguiente forma:

$$Flujo\ banda\ z = 1{,}1 * Flujo\ banda\ i + 3{,}15$$

En cuanto a los intervalos de confianza para la pendiente y la constante, estos son, respectivamente:

$$[0{,}94\colon 1{,}14], y\ [2{,}3\colon 7{,}86]$$

Los gráficos resultantes son los siguientes:

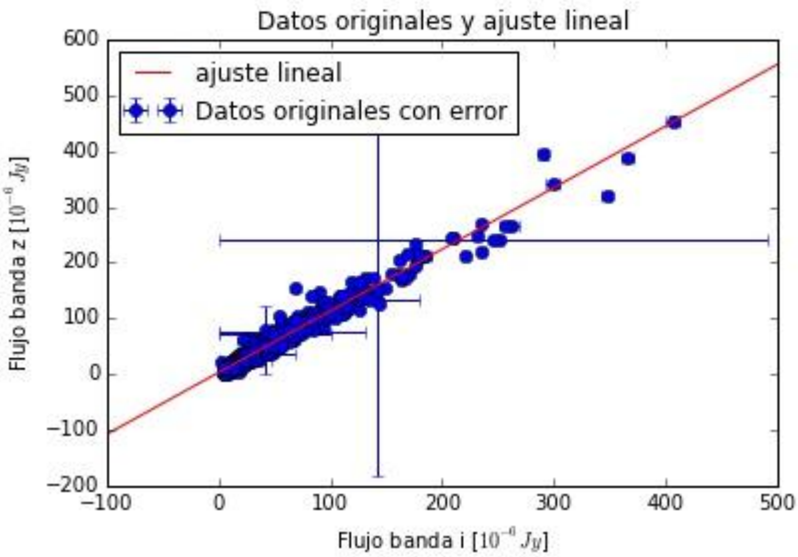

Fig 5: Gráfico de relación flujos de banda.

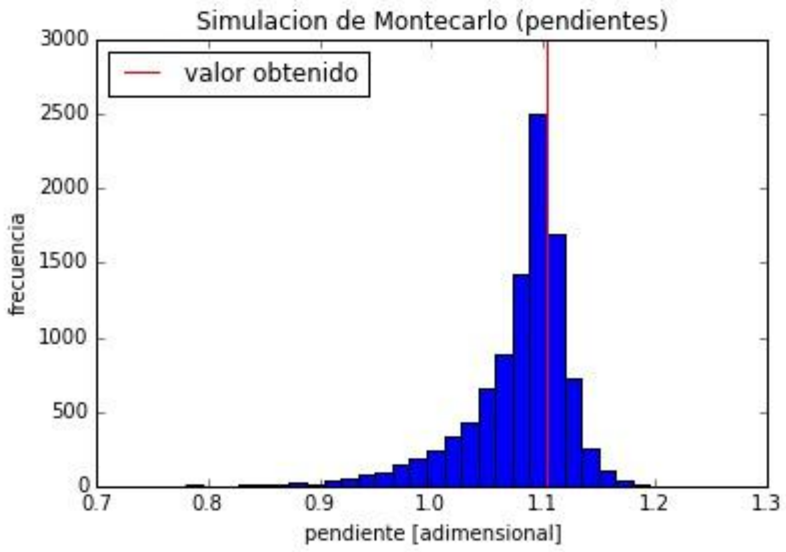

Fig 6: Histograma simulación Monte Carlo para pendiente.

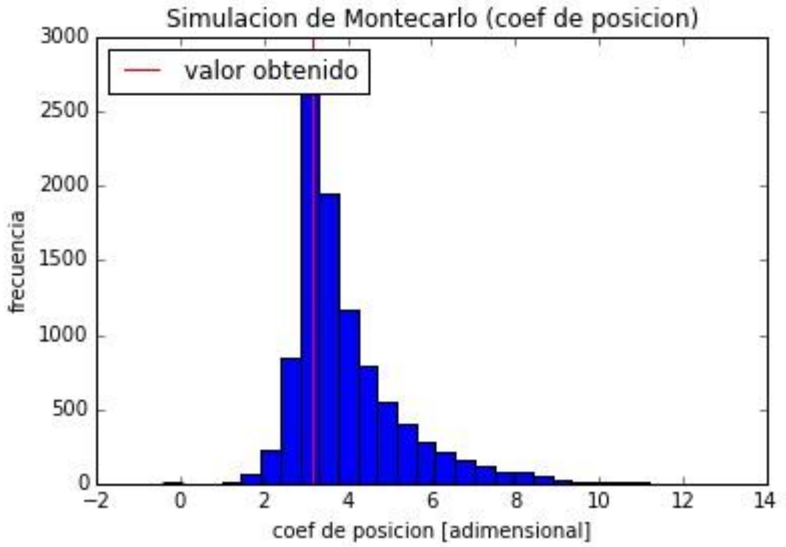

Fig 7: Histograma simulación Monte Carlo para constante.

4. Conclusiones

Los resultados de la P1 indican que, como H_0 es mayor a 1, que las galaxias más lejanas se alejan más rápido que las más cercanas. Esto implica que el universo está en expansión. Sin embargo, el valor de H_0 difiere de forma significativa entre las dos formas de definir la misma función. Esto tal vez se deba a los datos obtenidos por Hubble. La figura 1 muestra que éstos se encuentran muy dispersos, con varias galaxias *acercándose* a la Tierra (v negativo), incluso. Se puede apreciar (más o menos) la tendencia buscada, pero la relación no es precisa.

Los datos más exactos de la P2 llevan a un cálculo más preciso de H_0, que es bastante menor al cálculo original. Así, el universo se expande más lentamente que lo señalado por Hubble. La figura 3 muestra una clara tendencia lineal de los datos, y calcular H_0 no depende de la fórmula usada (la diferencia es despreciable, como se aprecia en el gráfico).

En la figura 5, se ve que la relación entre los flujos de onda tiene una clara tendencia lineal, y los errores asociados (salvo algunas excepciones) son despreciables en el gráfico.

Hay 2 conclusiones principales de esta tarea: Primero, se tiene que las funciones *curvefit* y *polyfit* funcionan muy bien para fitear las funciones dadas. Es más, *curvefit* también sirve bastante para relaciones lineales no polinomiales.

La segunda conclusión es que las simulaciones Bootstrap y Monte Carlo funcionan bien para calcular los intervalos de confianza. Los gráficos muestran que la mayoría de los datos simulados tienden hacia el dato real calculado. Además, se puede inferir que mientras más simulaciones se hagan, habrá una mayor tendencia hacia el dato real.

Todos los programas demoran un tiempo relativamente corto en ejecutarse.