

# Mínimos Cuadrados

Juan Manuel Cuellar Borrero

*Pontificia Universidad Javeriana Cali*

*Cali, Colombia*

[Juanma1909@javerianacali.edu.co](mailto:Juanma1909@javerianacali.edu.co)

## I.INTRODUCCION

Se desarrollará un archivo utilizando el lenguaje de programación Python, cumpliendo con los requisitos necesarios para dar solución al problema de mínimos cuadrados, a través de la implementación de los algoritmos de Ecuaciones Normales y de Transformaciones Householder.

## II.DESARROLLO DE CONTENIDO

### 1. *Materiales y Métodos*

#### 1. **Materiales**

- **Python:** Este fue el lenguaje utilizado para desarrollar los algoritmos de Ecuaciones Normales y Transformaciones Householder “Python es un lenguaje de programación de código abierto, orientado a objetos, muy simple y fácil de entender. Tiene una sintaxis sencilla que cuenta con una vasta biblioteca de herramientas, que hacen de Python un lenguaje de programación único.” [1].
- **Librería Numpy:** Esta librería ayudo a poder realizar todas las operaciones relacionadas con algebra lineal de forma rápida y efectiva. “ NumPy es un paquete de Python que significa “Numerical Python”, es la librería principal para la informática científica, proporciona potentes estructuras de datos, implementando matrices y matrices multidimensionales. Estas estructuras de datos garantizan cálculos eficientes con matrices. ” [2].
- **Librería Matplotlib:** Esta es la librería utilizada para poder graficar tanto los polinomios de ajuste como los puntos utilizados de las bases de datos. “Matplotlib es una librería de trazado utilizada para gráficos 2D en lenguaje de programación Python, es muy flexible y tiene muchos valores predeterminados incorporados que te ayudarán muchísimo en tu trabajo.” [3].
- **Datos Abiertos Colombia:** Esta es una página promovida por el gobierno colombiano, la cual pretende impulsar la transparencia y el acceso a la información pública.

- **Base de Datos No 1:** esta base de datos hace referencia a el numero de casos positivos de COVID-19 en el departamento del Meta en Colombia, este contiene 74 registros que van desde Marzo hasta Junio de 2020. Esta base fue tomada de Datos Abiertos Colombia

- **Base de Datos No 2:** Esta base de datos hace regencia al precio máximo del dólar por día, contiene 57 registros los cuales van desde Julio hasta Octubre del 2020. Esta Base fue tomada de Investing.com

#### 2. **Metodología**

Para la elaboración del algoritmo de ecuaciones normales, se hizo uso de la formula vista en clase:  $A^T A = L L^T$  donde A es la matriz construida a partir de los datos en cuestión y el número de parámetros del polinomio de ajuste a encontrar, y L hace referencia a la matriz resultante tras aplicar la descomposición de Cholesky. Tanto L como  $L^T$  son matrices triangulares las cuales pueden ser resueltas mediante la utilización de los métodos de *Sustitución sucesiva hacia atrás* y *“Sustitución sucesiva hacia adelante”*. Al finalizar se obtiene un arreglo con los coeficientes correspondientes al polinomio de ajuste.

Para la elaboración del algoritmo de Transformaciones Householder, se hizo uso del proceso visto en clase:  $H_n \dots H_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}$  donde R es una matriz triangular superior. Mediante transformaciones Householder sucesivas lo que se pretende es introducir ceros columna por columna por debajo de la diagonal. Para ello cada matriz  $H_x$  es construida de la siguiente forma:

$H_x = I - 2 \frac{v v^T}{v^T v}$ , donde el vector V se construye de acuerdo a la columna la que se le pretenda introducir ceros a través de esta transformación. Finalmente se utiliza Sustitución sucesiva hacia atrás para poder llegar a los coeficientes del polinomio de ajuste.

## 2. Resultados utilizando Base de datos No. 1

$|t| = 74, |y| = 74, g = 3$

### 1. Ecuaciones Normales:

- Error Medio: 16.3566
- Desviación Estándar: 31.4635
- Tiempo de ejecución: 0.007978

### 2. Transformaciones Householder:

- Error Medio: 16.3566
- Desviación Estándar: 31.4635
- Tiempo de ejecución: 0.004986

### 3. Análisis:

En primer lugar, se ve que tanto el error medio como la desviación estándar son prácticamente idénticos, digo prácticamente pues con apenas 4 cifras decimales es imposible notar el cambio, sin embargo si se usara una mayor cantidad de cifras decimales se vería que Householder da un error menor, sin embargo esta diferencia es casi imperceptible. Otro dato a analizar son los tiempos de ejecución donde se ve que el algoritmo de transformaciones Householder lleva la delantera con respecto a su rendimiento.

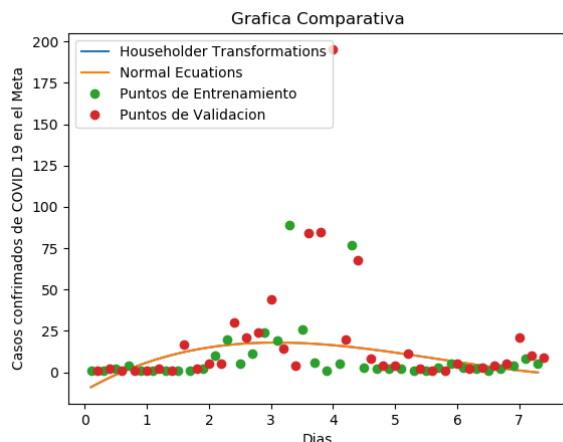


Fig. 1 Covid con ajuste de orden 3

$|t| = 74, |y| = 74, g = 6$

### 1. Ecuaciones Normales:

- Error Medio: 13.9444
- Desviación estándar: 29.7344
- Tiempo de ejecución: 0.038901

### 2. Transformaciones Householder:

- Error Medio: 13.9444
- Desviación Estándar: 29.7344
- Tiempo de ejecución: 0.009949

### 4. Análisis:

Nuevamente se ve que tanto los errores medios como la desviación estándar dan virtualmente iguales, sin embargo aquello que llama bastante la atención es el hecho de que ambos tiempos de

ejecución aumentaron considerablemente con respecto a lo que se presentó anteriormente.

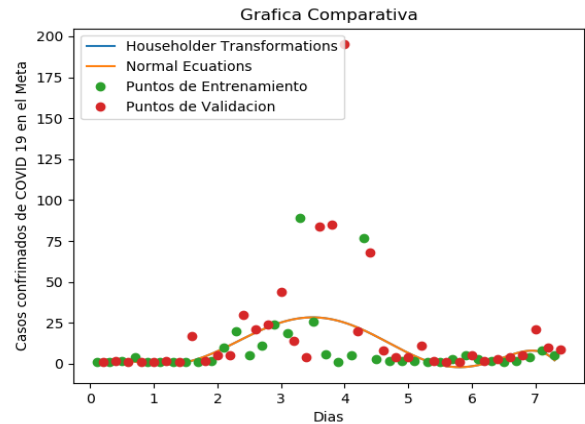


Fig. 2 Covid con ajuste de orden 6

$|t| = 74, |y| = 74, g = 12$

### 1. Ecuaciones Normales:

- Error Medio: 13.4597
- Desviación estándar: 30.1957
- Tiempo de ejecución: 0.021940

### 2. Transformaciones Householder:

- Error Medio: 13.6664
- Desviación Estándar: 30.2428
- Tiempo de Ejecucion: 0.010997

### 5. Análisis:

Por primera vez se vuelve perceptible una diferencia en cuanto a los errores medios y las desviaciones estándar de ambos algoritmos, esta diferencia no solo se hace visible en los datos mismos, sino también en la gráfica, pues hasta el momento, la gráfica de Transformaciones Householder parecía estar oculta tras la de Ecuaciones normales, sin embargo debido a esta diferencia entre los errores, ahora es un poco más visible la gráfica azul.

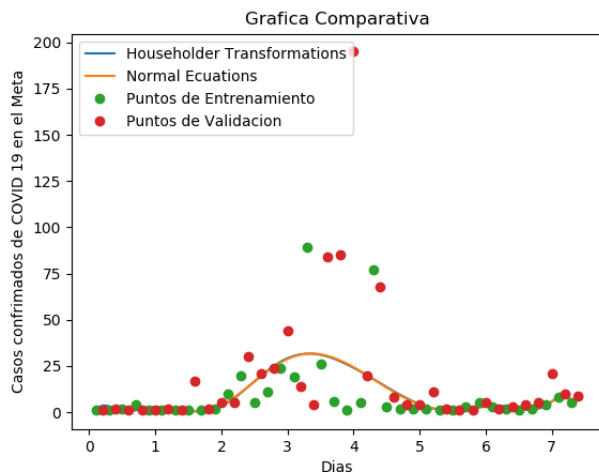


Fig. 3 Covid con ajuste orden 12

### 3. **Resultados Base de Datos No. 2**

$$|t| = 57, |y| = 57, g = 3$$

#### 1. **Ecuaciones Normales:**

- Error Medio: 33.6238
- Desviación estándar: 27.9714
- Tiempo de ejecución: 0.007978

#### 2. **Transformaciones Householder:**

- Error Medio: 36.6238
- Desviación estándar: 27.9714
- Tiempo de ejecución: 0.003989

#### ➤ **Análisis:**

Nuevamente se ve que tanto el error medio como la desviación estándar de ambos algoritmos son virtualmente los mismos si se toma únicamente 4 cifras decimales, de igual forma se ve de nuevo reflejado que el rendimiento del algoritmo de Householder frente al de Ecuaciones Normales, es mucho mejor.

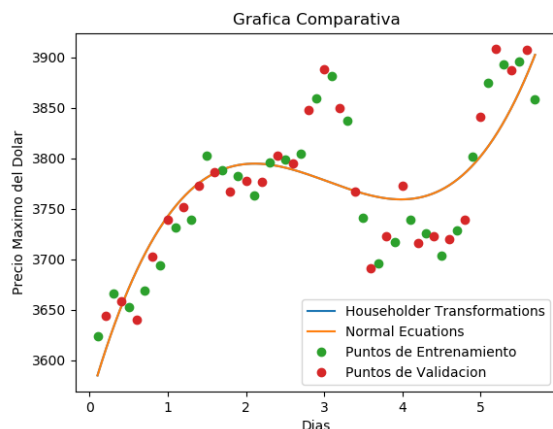


Fig. 4 Dólar con ajuste orden 3

$$|t| = 57, |y| = 57, g = 6$$

#### 1. **Ecuaciones Normales:**

- Error Medio: 27.9229
- Desviación estándar: 18.5556
- Tiempo de ejecución: 0.009945

#### 2. **Transformaciones Householder:**

- Error Medio: 27.9229
- Desviación estándar: 18.5556
- Tiempo de ejecución: 0.004986

#### ➤ **Análisis:**

Al igual que con el caso anterior, no existe una diferencia notable en cuanto los errores medios o las desviaciones estándar. Cabe la pena resaltar el hecho de a diferencia de la base de datos anterior, aquí hay un salto fuerte en los errores medio de un orden de polinomio a otro.

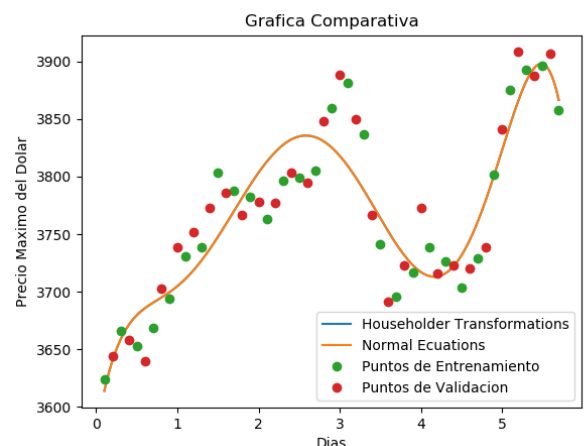


Fig. 5 Dólar con ajuste orden 6

$$|t| = 57, |y| = 57, g = 13$$

#### 3. **Ecuaciones Normales:**

- Error Medio: 30.8808
- Desviación estándar: 24.1915
- Tiempo de ejecución: 0.010998

#### 4. **Transformaciones Householder:**

- Error Medio: 20.6075
- Desviación estándar: 18.9090
- Tiempo de ejecución: 0.005013

#### ➤ **Análisis:**

Es en este orden de polinomio, que se ve una clara diferencia en cuanto a errores y desviaciones estándar en ambos algoritmos, es curioso notar que el algoritmo de ecuaciones normales da un mayor error medio con un orden 13 que con un orden 6, y pasa lo mismo con respecto a su desviación estándar. Se puede notar por lo datos arrojados que el algoritmo de Householder trabaja mejor una vez se llega a este punto. Además de esto se ve

gráficamente una clara diferencia en cuanto a los polinomios de ajuste arrojados por cada algoritmo.

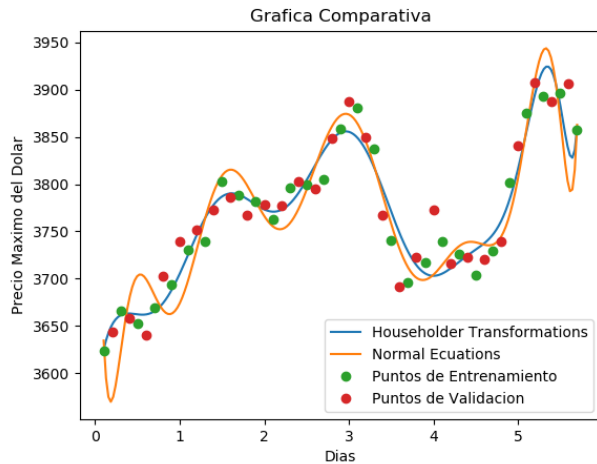


Fig. 6 Dólar con ajuste orden 13

### III. CONCLUSIONES

Primeramente, se ve una clara diferencia en cuanto rendimiento por parte del algoritmo de Transformaciones Householder, pues en cada una de las pruebas su tiempo de ejecución era menor que el Ecuaciones nomrales, se cree que esto puede ser debido a que la descomposición de Cholesky puede llegar a ser muy costosa a nivel computacional. De igual manera, Householder demuestra su dominancia debido a que a partir de orden 13, para la Base de datos No 1, y 14, para la Base de datos No. 2, ecuaciones normales (por lo menos para estas bases de datos) deja de servir, arrojando un error que dice que la matriz A no es definida positiva, por lo que no es posible resolverlo utilizando Ecuaciones Normales.

Con respecto a las bases de datos, fue necesario realizar unas transformaciones para que estas funcionaran, en general lo que se hizo fue en de tener día 1, 2, 3..., se transformó para que fuera día 0.1, 0.2, 0.3, ... Esto porque había casos en la que la gráfica se iba a infinito.

### IV. REFERENCIAS

- [1] A. Soloaga (2018, Octubre 19) "Principales Usos dePython".[Online]. Disponible en : <https://www.akademus.es/blog/programacion/principales-usos-python/>

- [2]. L. González. (2018, Septiembre 21) "Introducción a la librería NumPy de Python – Parte 1".[Online]. Disponible en : <https://ligdigonzalez.com/introduccion-a-numpy-python-1/>

- [3]. L. González. (2018, Octubre 19). "Introducción a la Librería Matplotlib de Python – Parte 1".[Online]. Disponible en : <https://ligdigonzalez.com/libreria-pandas-de-matplotlib-tutorial/>

- [4] Gov.co, Datos Abiertos [Online]. Disponible en: <https://www.datos.gov.co/>

- [5] Investing.com, USD/COP - Dólar Peso colombiano [Online] Disponible en: <https://es.investing.com/currencies/usd-cop-historical-data>