**Ajustes Polinomiales**

Autor: Daniel Felipe Ladino Velásquez

Jonatan Hernández Henao

*IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: d.ladino@utp.edu.co

Correo-e: jonatan.hernandez@utp.edu.co

**Introducción**

Hemos estado conviviendo con el covid 19 desde hace 6 meses, lo que se busca con este programa es modelar estadísticamente los casos nuevos cada día de los meses junio, julio y agosto. Teniendo en cuenta si los contagios son crecientes o decrecientes, buscamos hallar una predicción con el fin de saber qué día llegaremos al número de contagios que nosotros especificamos que queríamos llegar exactamente. Esto lo tomamos con el número de contagios por día y lo dividimos en semanas al momento de modelar, quedando una gráfica de número de contagios vs semanas. En nuestros modelados el número de contagios llega a un pico y tiende a disminuir los contagios por día, por tal razón nuestra meta en la predicción es menor a la del pico.

Palabras clave

* Estadística
* Modelado
* Contagios
* Predicción
* Búsqueda

**Introduction**

We have been living with covid 19 for 6 months, what is sought with this program is to statistically model new cases every day in June, July and August. Taking into account whether the infections are increasing or decreasing, we seek to find a prediction in order to know which day we will reach the number of infections that we specify that we want to arrive exactly. We take this with the number of infections per day and divide it into weeks at the time of modeling, leaving a graph of the number of infections vs. weeks. In our models, the number of infections reaches a peak and tends to decrease infections per day, for this reason our goal in the prediction is lower than that of the peak.

Keywords

• Statistics

• Modeling

• Contagions

• Prediction

• Search

En primer lugar, se importa la librería import os que es la del sistema operativo y a su vez para utils importamos from utils import DATA\_DIR, CHART\_DIR que nos darán los directorios tanto de los datos creados como los gráficos modelados.

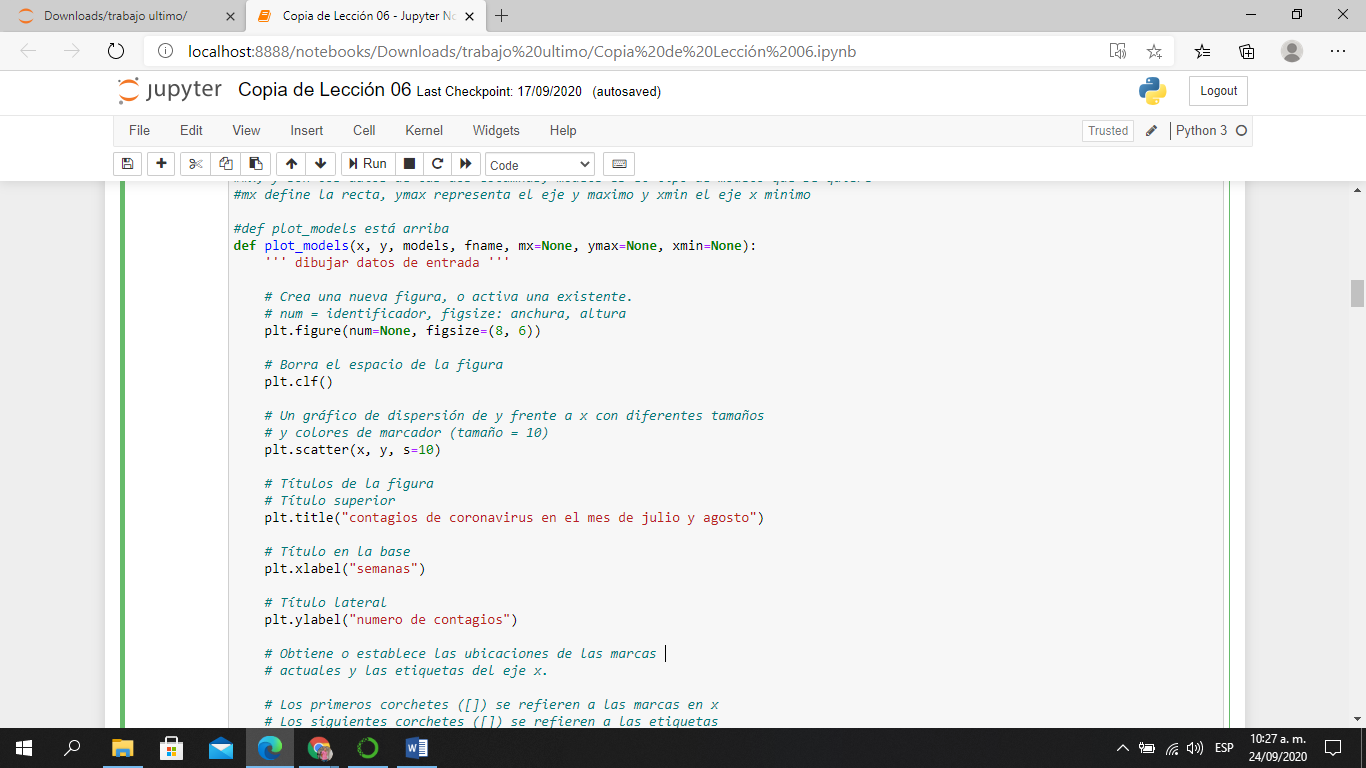
Importamos numpy asignándole un alias np, también importamos las librerías scipy y matplotliblas cuales nos ayudaran a organizar en subpaquetes dejando a cada uno enfocado o dirigido a unos cálculos específicos permitiéndonos modelar de una manera mucho más correcta, a la librería scipy también se le asigna un alias que es sp.

data = np.genfromtxt(os.path.join(DATA\_DIR, "estadisticas.tsv"), delimiter="\t") con esta función llamamos nuestros datos guardados como texto para darles un orden y hacerle su respectiva tabulación

definimos los colores a tomar con g = green, k = black, b = blue, m = magenta, r = red # g = verde, k = negro, b = azul, m = magenta, r = rojo colors = ['g', 'k', 'b', 'm', 'r'] para tener una mejor organización al momento de modelar nuestras graficas al igual definimos el tipo de línea que vamos a utilizar para cada modelado y eso se logra con la función linestyles = ['-', '-.', '--', ':', '-'].

Se crea un vector x el cual corresponde a la primera columna de nuestros datos y un vector y que corresponderá a la segunda columna de nuestros datos.

print("Número de entradas incorrectas:", np.sum(np.isnan(y)) con esta función buscamos hallar las imperfecciones o datos erróneos de nuestra lista de datos



Con estas instrucciones buscamos definir y crear el modelado donde se plasma la información que lleva este grafico tal como: el ancho, la altura, la dispersión de los puntos en la gráfica y a su vez definimos x y y con sus respectivos nombres que serían semanas y número de contagios.

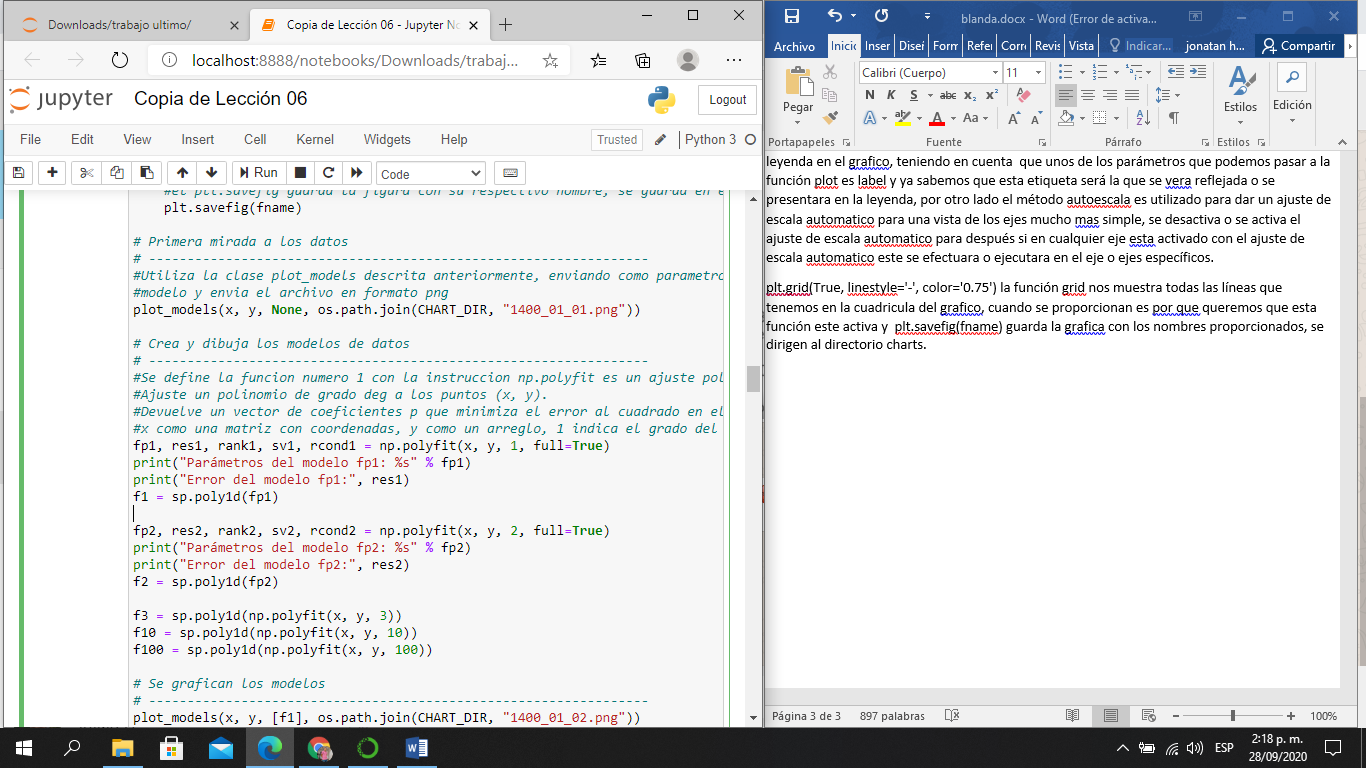
[w \* 7 \* 1 for w in range(9)], ['semana %i' % w for w in range(9)]) en esta función lo que hacemos es una conversión buscando volver todos los datos hallados que tenemos en días a semanas dándole un rango de 13 que serán la cantidad de semanas de los tres meses.

Con la función linspace lo que buscamos es devolver números especificados uniformemente, pero en un intervalo especificado, mirándolo en el caso que tenemos lo estamos ejecutando o usando la función linspace para el conjunto de valores establecidos.

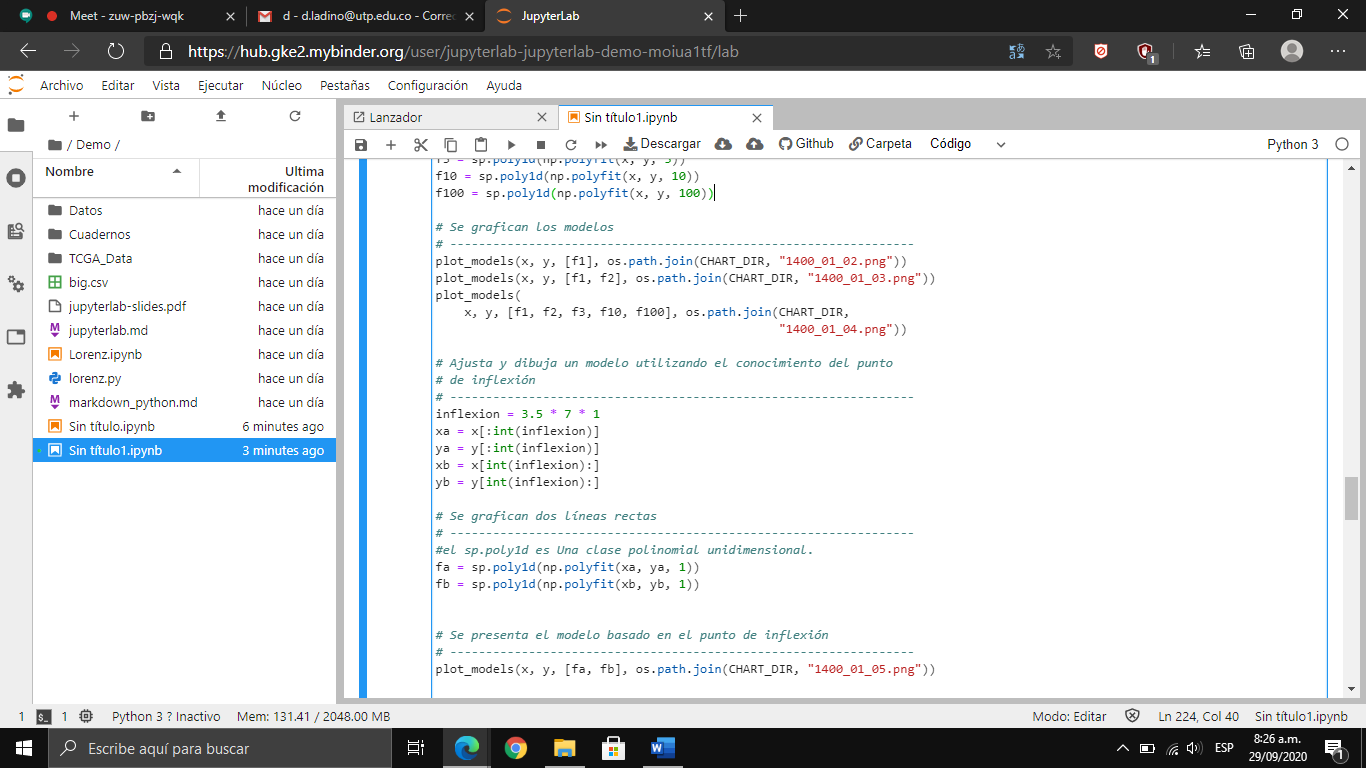
La función que toma zip () es coger elementos que pueden variar de cero o muchos más elementos para agregarlos en una tupla para luego devolverlos, este proceso se repite una y muchas veces generando un ciclo. plt.plot(mx, model(mx), linestyle=style, linewidth=2, c=color) con esta función lo que buscamos es como darle el tipo o estilo de línea, el color etc. En pocas palabras con esta función personalizamos la gráfica o el modelado que estamos elaborando o queremos realizar.

plt.legend este método de un determinado conjunto de ejes como lo es x y y nos mostrara la leyenda en el gráfico, teniendo en cuenta que unos de los parámetros que podemos pasar a la función plot es label y ya sabemos que esta etiqueta será la que se verá reflejada o se presentara en la leyenda, por otro lado el método autoescala es utilizado para dar un ajuste de escala automático para una vista de los ejes mucho más simple, se desactiva o se activa el ajuste de escala automático para después si en cualquier eje esta activado con el ajuste de escala automático este se efectuara o ejecutara en el eje o ejes específicos.

plt.grid(True, linestyle='-', color='0.75') la función grid nos muestra todas las líneas que tenemos en la cuadricula del gráfico, cuando se proporcionan es porque queremos que esta función este activa y plt.savefig(fname) guarda la gráfica con los nombres proporcionados, se dirigen al directorio charts.



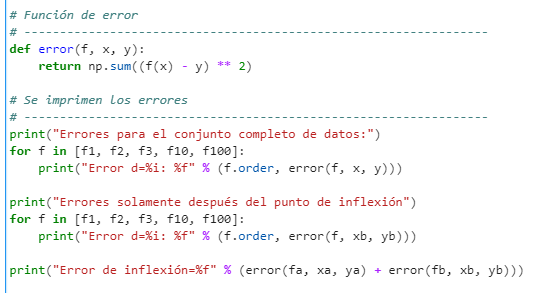
Lo que hacemos en las funciones anteriores es crear y plasmar los modelos de datos donde se define la primera función con una instrucción llamada np.polyfit que es un ajuste polinomial de los mínimos cuadrado, ajustamos el polinomio a los puntos x y y, y nos devolverá un vector con coeficiente p que lo que hace en la función es minimizar el error al cuadrado.



Lo que se hace con esta función es darle vida al modelo que queremos graficar usando la función plot\_models.

Se busca el punto de inflexión que es donde toma el pico de contagios por día y empieza a descender apuntando hacia la predicción dada más adelante con el fin de tener los datos finales más acertados.

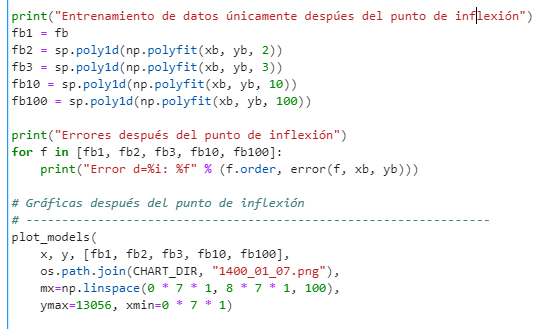
Se traza dos líneas rectas con la función fa = sp.poly1d(np.polyfit(xa, ya, 1)) que es una clase polinomial unidimensional que es de conveniencia y es utilizada para encapsular operaciones naturales en polinomios para que dichas operaciones puedan tomar su forma habitual en el código.



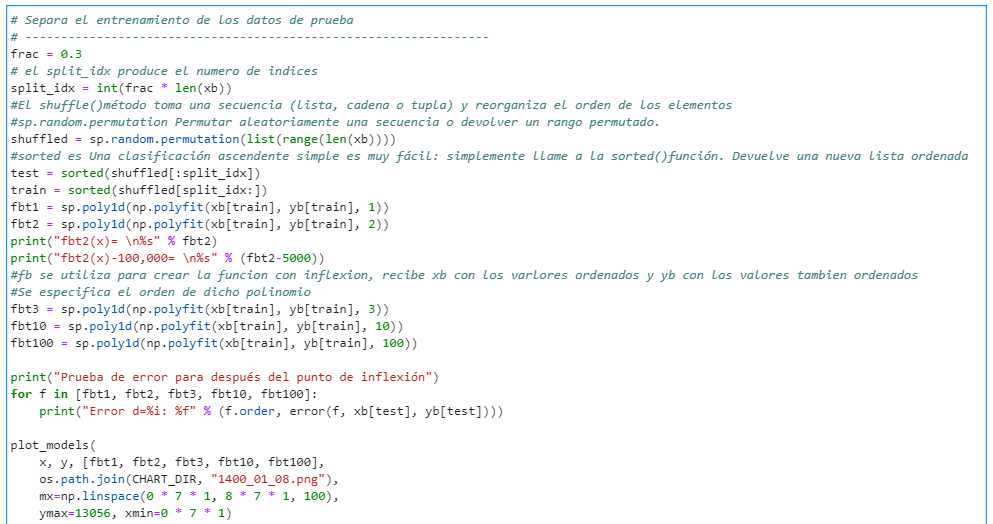
Lo que hace es mostrarnos en pantalla los errores de cada una de las gráficas ya sea para el conjunto completo de los datos o del punto de inflexión.

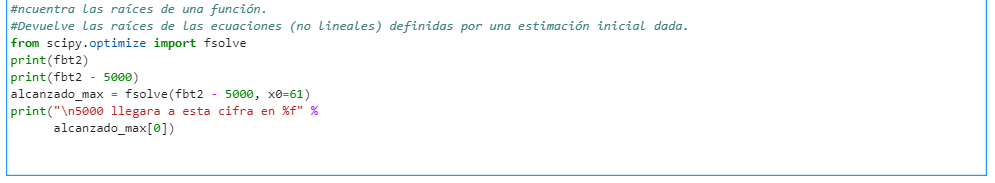
lo que se hace aquí es que se indaga en los datos, luego se introducen al código y este hace una especie de hipótesis a partir de los datos suministrados.

Con os.path.join se unen varios componentes de ruta pero teniendo en cuenta que llevara exactamente un separador de directorio, después de cada parte no vacía habiendo un componente de ruta al final, se regresa los números espaciados uniformemente en un intervalo que se ha especificado y también mostramos o devolvemos el número generado de las muestras espaciadas uniformemente las cuales se calcularon en el intervalo especificado.



El entrenamiento de datos lo que hace es utilizar numpy para entrenar la variable fb2 y de esta manera el sistema determina el punto de inflexión gracias a las variables xb y yb, estas variables fueron definidas anteriormente estableciendo un punto de inflexión y así dividir los datos para diferenciar los más relevantes. El número entero 2 indica el grado del polinomio con el que se va a hacer la regresión lineal.

El sp.poly es una función para graficar y recibe como argumento el entrenamiento de numpy, se crean otras tres variables modificando solamente el grado de polinomio desde 3 hasta 100.



Con estos códigos lo que se busca es que en cada recta (x, y) se grafiquen cada uno de los valores que previamente se añadieron, esto con el fin de que se puedan ver gráficamente en forma ascendente y con cierto rango para poder visualizarlo en este, se hace una predicción con base a los datos que se tienen y dicha predicción da como finalidad tener un poco de certeza acerca de cómo estarán los contagios más adelante y si estuvo cerca o no del valor que le demos a esta predicción.