MODELADO MATEMÁTICO 2
Introducción a la estadística bayesiana

Carlos Andrés Bautista Torres

Aspectos Generales

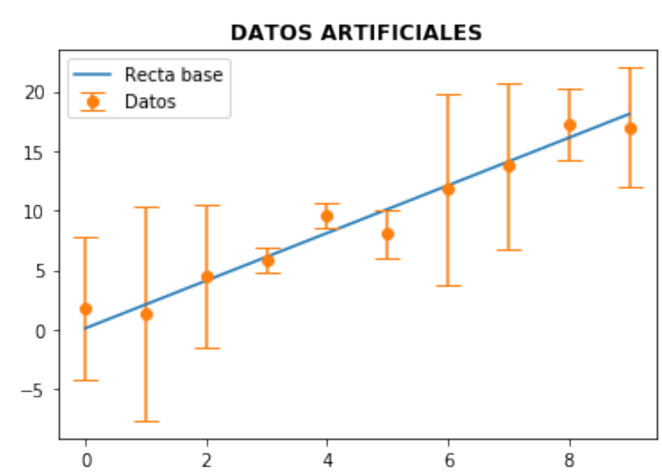
Este notebook da una introducción a la estadística bayesiana mediante el modelamiento de una serie de puntos a partir de la ecuación de una recta. Consta de cinco partes adicionales a las descritas en la primera entrega del trabajo, dichas partes son: 1) hallar la desviación estándar en el error de los parámetros. 2) realizar los contornos de probabilidad para 66%, 90%, 95% y 99%. 3) crear una malla adaptativa cuya región central tenga mejor resolución. 4) hacer un análisis teórico de la elipse que representa la distribución de probabilidad. 5) incorporar un prior de forma escalón unitario, que sólo admita valores positivos para el intercepto, $b \geq 0$.

Estructura del Análisis

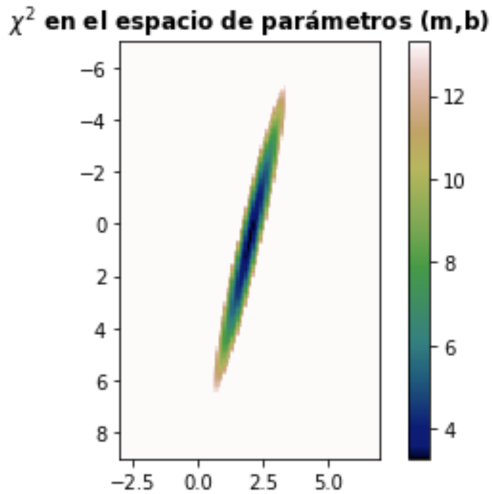
Esta carpeta consta de un archivo .ipynb, en donde se encuentra el análisis previamente explicado.

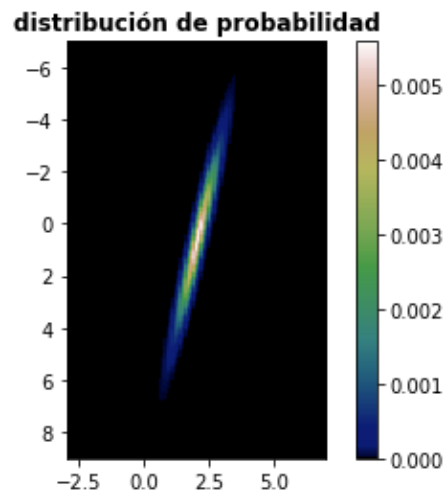
CREACIÓN DE DATOS:

Se define una recta $y = mx + b$ con $m = 2$ y $b = 0.1$. Luego se les agrega ruido y unas barras de error, obteniendo el siguiente resultado

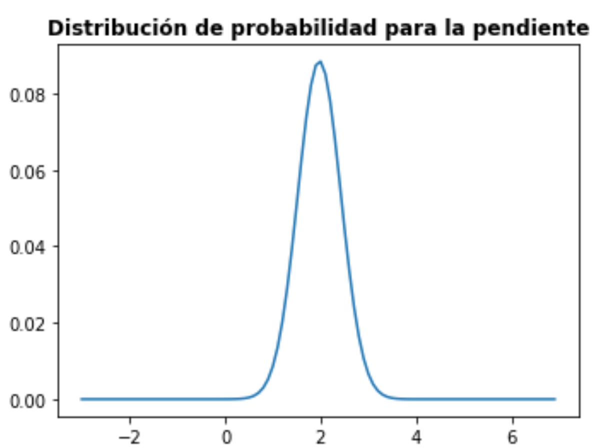


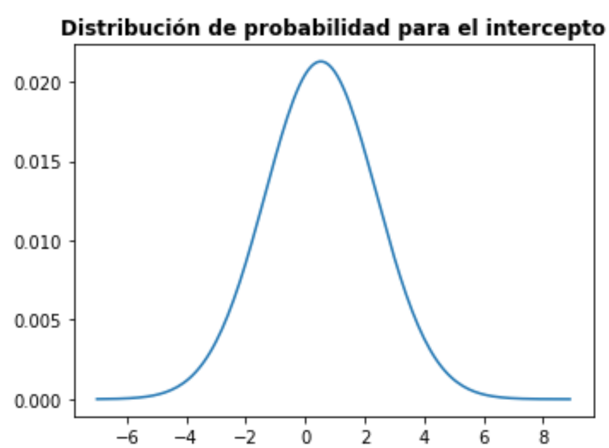
Mediante una función de optimización se estima que los mejores valores para m y b son 1.8812463680832587 y 0.6126560864239234 respectivamente. A partir de los dos valores anteriores se crea una malla ***con buena resolución*** en el espacio de parámetros (m, b) para observar el comportamiento del χ^2. Y a su vez, se puede usar la ecuación $P\left(x_k | \mu, \sigma, H\right) \propto \exp\left[-\frac{(x_k - \mu)^2}{2\sigma^2}\right]$ para determinar la distribución de probabilidad. Dando como resultado





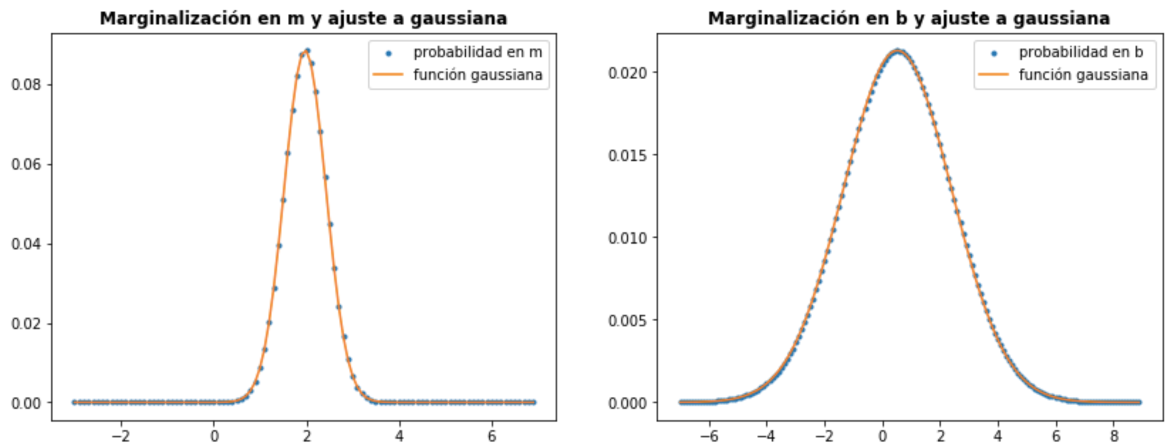
Realizando una marginalización a partir de la siguiente fórmula $P(x|H) = \int P(x, y|H)dy$, se determinan las distribución de probabilidad tanto para la pendiente como para el intercepto.





DESVIACIÓN ESTANDAR DE LOS PARÁMETROS:

Debido a que los errores provienen de una distribución gaussiana, se utiliza la función "Make_gaussian" para encontrar la amplitud, dispersión y la media de la función gaussiana que mejor describe el conjunto de datos. Este proceso de optimización es similar al que se ha llevado a cabo en los trabajos previos, y da como resultado la siguiente gráfica,

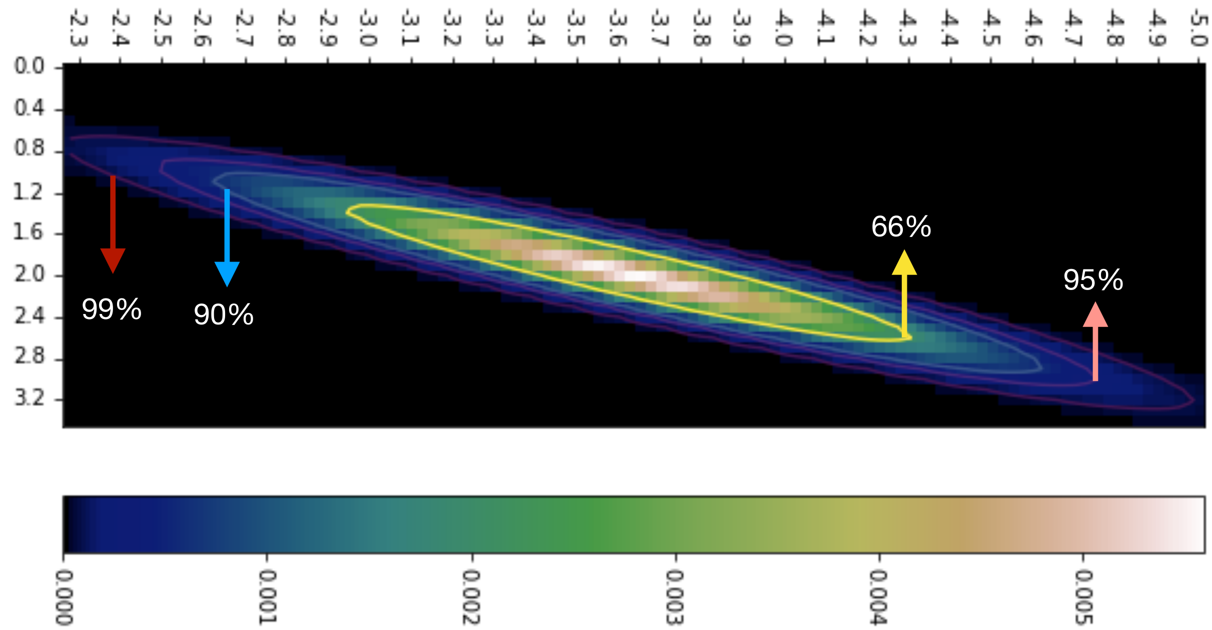


con unos parámetros de dispersión $\sigma_m = 0.4507494$ y $\sigma_b = 1.8737289$ y un valor medio de $\mu_m = 1.97394727$ y $\mu_b = 0.52719441$.

CONTORNOS DE PROBABILIDAD:

Para obtener los contornos de probabilidad de 66%, 90%, 95% y 99%, se debe sumar la probabilidad de cada celda dentro del contorno, esto se hace mediante una función llamada *mask_finder.*

```
def mask_finder(limit):
   prob = 0 # Se inicializa el contador.
   for i in range(b_list3.size):
      for j in range(m_list3.size):
         if  like3[i,j] >= limit:
            prob = prob + like3[i,j] # Esto se puede debido a que la malla no es adaptativa.
   return prob
```

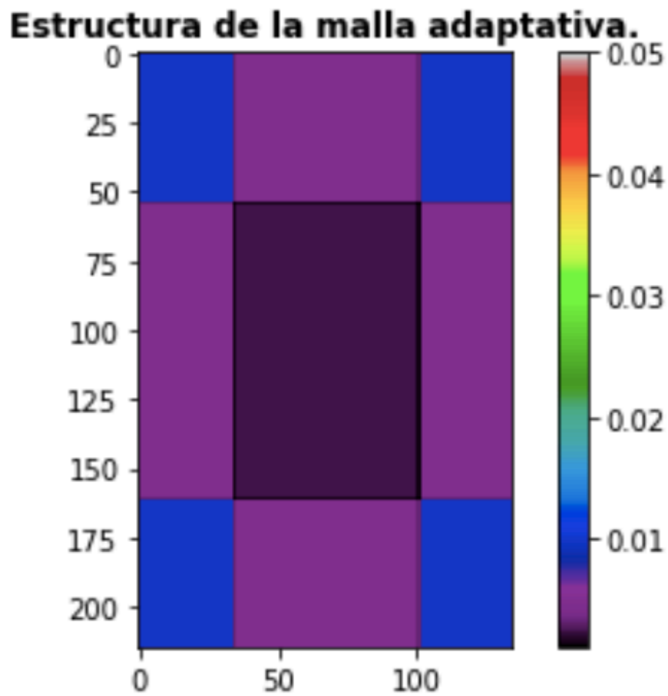


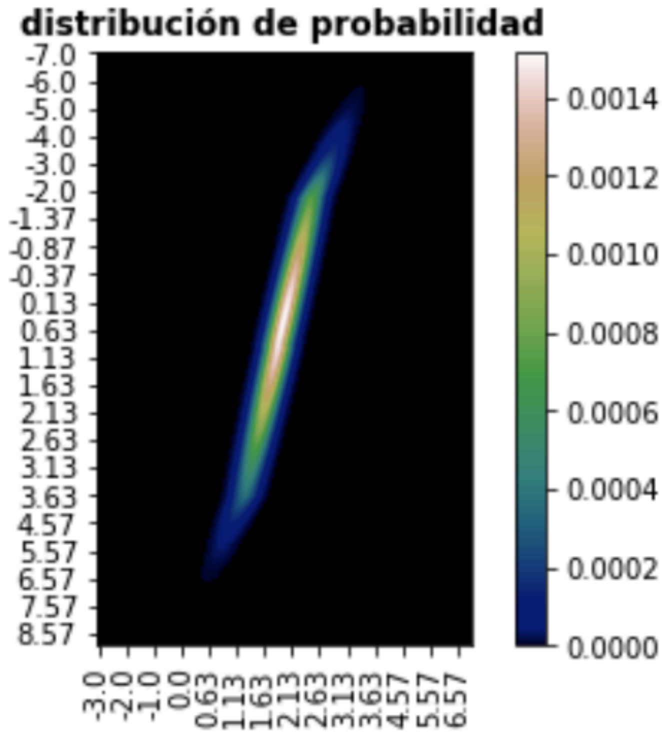
MALLA ADAPTATIVA:

Para esta sección se creó un "*np.arange*" adaptativo, cuyo espaciamiento en el rango de números centrales sea mayor. La función que se definió para lograr esto es la siguiente:

```
def Adapt_mesh(xmin, xmax, delta1, delta2):
    rango = xmax - xmin # Determina el intervalo total de la malla
    x1 = np.arange(xmin, xmin + rango/3, delta1)
    x2 = np.arange(xmin + rango/3, xmin + 2*rango/3 , delta2)
    x3 = np.arange(xmin + 2*rango/3, xmax, delta1)
    x  = np.concatenate((x1, x2, x3), axis=None)
    return x
```

La función pide un rango (xmin, xmax) y dos deltas de espaciamiento asociadas a dos resoluciones distintas. Posteriormente divide el np.arange en tres intervalos. En los intervalos extremos utiliza la delta asociada con una menor resolución, y en el intervalo central utiliza la delta de mayor resolución. Al crear el espacio de parámetros la malla queda de la forma como se muestra en la siguiente figura a la derecha. El inconveniente con este método es que refina un poco cuatro zonas en donde no debería, pero dicha resolución es intermedia, es decir, la región central sigue siendo la zona de mayor resolución. A la derecha se tiene la distribución de probabilidad obtenida para la maya adaptativa. Los parámetros $\sigma_m, \sigma_b, \mu_m, \mu_b$ son prácticamente los mismos que en el caso anterior. $\sigma_m = 0.40899289$, $\sigma_b = 1.87372641$, $\mu_m = 1.95410177$, $\mu_b = 0.52719453$.





ANÁLISIS TEÓRICO DE LA ELIPSE:

La matriz de Fisser o Hessiano se calcula a partir de las segundas derivadas de la probabilidad con respecto a los parámetros.

$$\mathbf{F} = -\begin{bmatrix} A & C \\ C & B \end{bmatrix}, \text{ en donde } A = \left(\frac{\partial^2 L}{\partial m^2}\right)_0, \quad B = \left(\frac{\partial^2 L}{\partial b^2}\right)_0, \quad C = \left(\frac{\partial^2 L}{\partial m \partial b}\right)_0.$$

Teniendo en cuenta que la distribución de probabilidad es de la forma

$$P \propto \mathbf{exp}\left[-\left(\frac{(m-m_0)^2}{2\sigma_m^2} + \frac{(b-b_0)^2}{2\sigma_b^2}\right)\right], \text{ su logaritmo natural es}$$

$$L = \mathbf{ln}\, P = \left[-\left(\frac{(m-m_0)^2}{2\sigma_m^2} + \frac{(b-b_0)^2}{2\sigma_b^2}\right)\right], \text{ y por lo tanto, las primeras derivadas}$$

con respecto a los parámetros son

$\dfrac{dL}{dm} = -\dfrac{m - m_0}{\sigma_m^2} \quad \wedge \quad \dfrac{dL}{db} = -\dfrac{b - b_0}{\sigma_b^2}$, y sus segundas derivadas son

$\dfrac{d^2L}{dm^2} = -\dfrac{1}{\sigma_m^2} \quad \wedge \quad \dfrac{d^2L}{db^2} = -\dfrac{1}{\sigma_b^2}$, siendo las derivadas mixtas nulas y la matriz de

Fisser adquiere la forma

$$\mathbf{F} = -\begin{bmatrix} \dfrac{1}{\sigma_m^2} & 0 \\ 0 & \dfrac{1}{\sigma_b^2} \end{bmatrix} = -\begin{bmatrix} 4.9229 & 0 \\ 0 & 0.2848 \end{bmatrix}$$

con autovalores $\lambda_1 = -4.9229$ y $\lambda_2 = -0.2848$, al ser una matriz diagonal. Para hallar los valores de los semi-ejes de la elipse es necesario realizar el siguiente producto:

$$Q = k = \mathbf{W}\ \mathbf{F}\ \mathbf{W}^T = [\mu_m \ \ \mu_b] \begin{bmatrix} \dfrac{-1}{\sigma_m^2} & 0 \\ 0 & \dfrac{-1}{\sigma_b^2} \end{bmatrix} \begin{bmatrix} \mu_m \\ \mu_b \end{bmatrix} = -19.2742 \ ,$$

por lo tanto los valores para los semi-ejes son:

$$e_1 = \sqrt{\frac{k}{\lambda_1}} = 1.9787 \ \ \wedge \ \ e_2 = \sqrt{\frac{k}{\lambda_2}} = 8.2265$$

Esto se puede verificar usando las siguientes expresiones:

$\sigma_m = \sqrt{\dfrac{-B}{AB - C^2}} \quad \wedge \quad \sigma_b = \sqrt{\dfrac{-A}{AB - C^2}}$ con $C = 0$ se tiene que

$\sigma_m = \sqrt{\dfrac{-1}{A}} \quad \wedge \quad \sigma_b = \sqrt{\dfrac{-1}{B}}$ siendo $A = \dfrac{-1}{\sigma_m}$ y $B = \dfrac{-1}{\sigma_b}$.

PRIOR:

Si se considerara que los valores del intercepto tienen que ser siempre positivos, la manera de expresar esta condición es usando un prior de forma de escalón unitario, $P(H) = 0$ **si** $b < 0$ y $P(H) = 1$ **si** $b \geq 0$.

```
def prior(b):
    if b < 0:
        pri = 0
    else:
        pri = 1
    return pri
```

Al imponer el prior se re-distribuye la función de probabilidad hacia la zona de valores $b >= 0$, sin embargo la forma de la elipse no se ve modificada como tal, sino cortada abruptamente por el eje $b = 0$.