Interrogación 3

Rodrigo Carvajal Pizarro (rcarvaja@astro.puc.cl)
`https://github.com/racarvajal/Interrogacion_3_Astro_Stats.git`

Resolución

Problema 1

Los datos a de las observaciones pueden verse en la Figura 1. En ella, se graficó el logaritmo de los valores entregados. Se puede notar que no existe una diferencia demasiado notoria entre las observaciones que no cuentan con un tránsito y aquella que sí lo incluye en su curva.

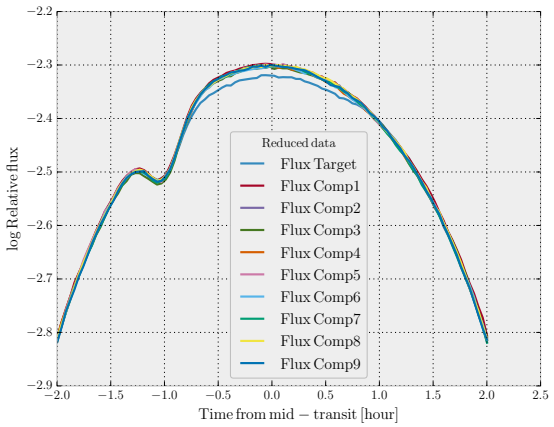

Figura 1: Logaritmo de datos entregados

a) Para obtener las componentes principales de los datos entregados, se utilizó el módulo `mlab` de `matplotlib`. Se le entregan los datos de las observaciones sin tránsitos y éste calcula las componentes, sus pesos (valores propios de la matriz de covarianza) y las fracciones dentro del total acumulado.

   El resultado de este proceso puede verse en los gráficos de la Figura 2. Sin mucha dificultad, puede notarse que la primera componente es la que más influencia tiene en el flujo.

   De los gráficos de la Figura 3, puede verse, numéricamente, que la primera componente es la que más peso tiene sobre las otras. Casi la mayor parte del peso total de las componentes es acumulado por una sola.

   Tan solo con los resultados del Análisis de Componentes Principales, puede adelantarse que, tomando solo la primera de las componentes, el modelo no pierde información. Por lo tanto, se puede, eventualmente, trabajar con una sola componente, la primera.

b) Si se incorpora un modelo de tránsito y una constante aditiva al modelo completo, es posible obtener parámetros más acertados. Pero, también, se corre el riesgo que obtener una combinación poco realista de éstos. Por lo tanto, es necesario establecer el número adecuado de parámetros que no sobre ni subestime los datos.

   Como hay una alta nolinealidad del modelo en los parámetros, se utlizará el algoritmo MCMC implementado a través del módulo `emcee` en Python.

   Este módulo requiere la definición de las funciones de verosimilitud, priors y probabilidad posterior del modelo y sus parámetros asociados. Se asume que el error asociado (también, un parámetro a estimar) es gaussiano e independiente.

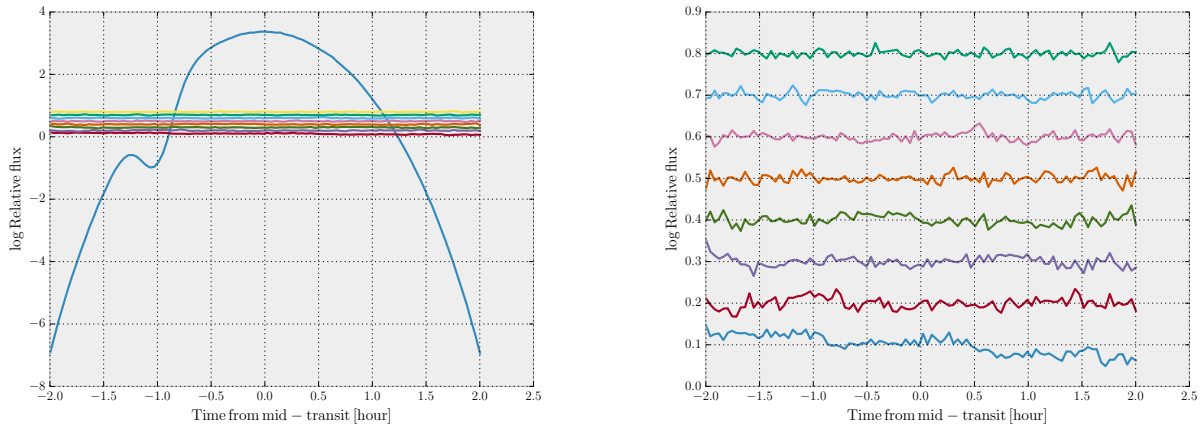

Figura 2: Componentes Principales calculadas (Cada componente está desplazada verticalmente para notar su estructura). Izquierda: todas las componentes. Derecha: todas las componentes salvo la primera

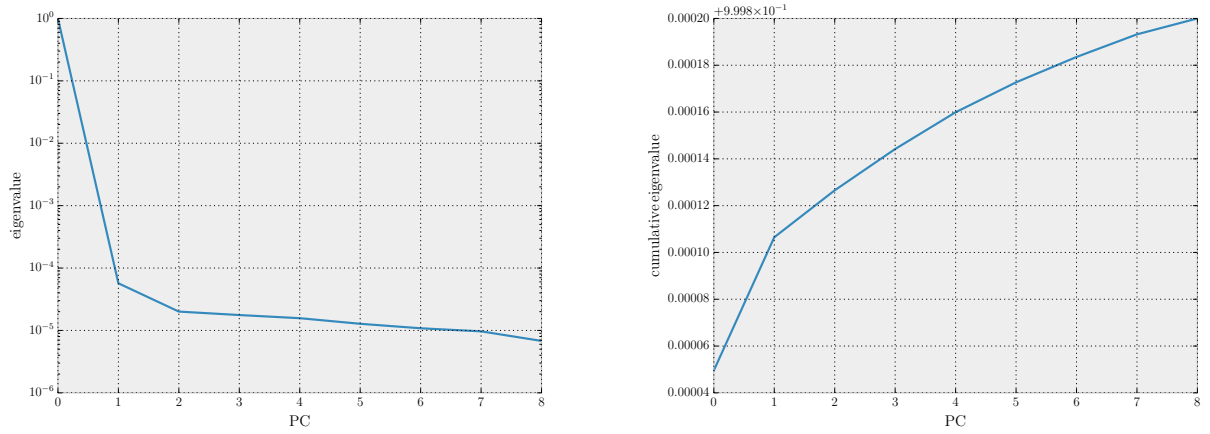

Figura 3: Pesos de las componentes obtenidas. Izquierda: Peso de cada una. Derecha: peso acumulado

Observando los valores originales de los datos y teniendo ciertas consideraciones físicas asociadas al modelo buscado, se utilizaron priors uniformes en los siguientes intervalos.

$$-3 < c < 0$$

$$0 < rp < 2$$

$$0 < a < 100$$

$$0 < inc < 90$$

$$10^{-5} < \sigma < 10^{-3}$$

$$0 < \alpha_i < 1{,}5 \times \text{pca}_i$$

Se utilizaron 200 *walkers*. La etapa de *burn-in* de MCMC fue ejecutada en 1000 iteraciones y la de MCMC efectiva, constó de 3500 iteraciones.

Para interpretar cada ejecución de MCMC con diferente número de componentes de PCA utilizadas, se graficó una comparación entre los datos originales y un subconjunto de iteraciones del algoritmo.

También, se graficaron los datos originales de la observación con tránsito junto con el valor medio de todas las iteraciones de MCMC. Los gráficos mencionados para 1, 5 y 9 componentes pueden verse en las Figuras 4, 5 y 6.

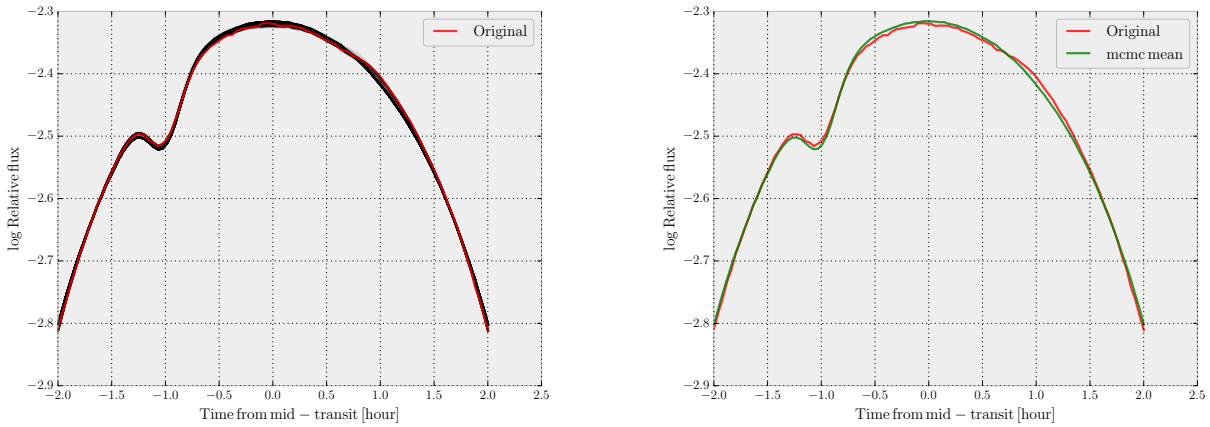

Figura 4: Resultado de MCMC para 1 componente de PCA considerada

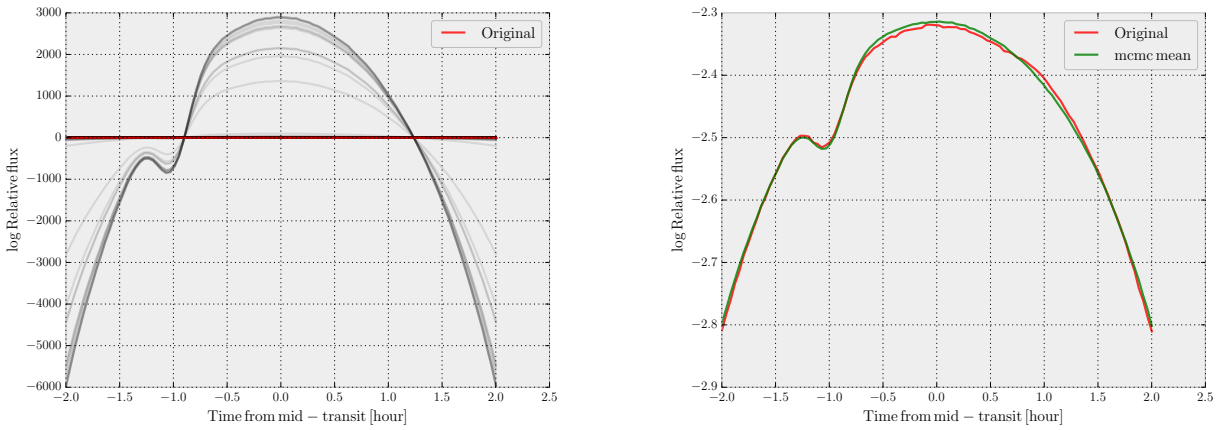

Figura 5: Resultado de MCMC para 5 componentes de PCA consideradas

De las figuras, puede notarse que el ajuste con solo una componente es bastante adecuado. Ya con cinco componentes, el valor medio sigue siendo bueno. Pero ya hay más ejecuciones que se alejaron bastante de los datos originales. Y, para nueve componentes, es notorio que el ajuste con MCMC no es completamente adecuado; muchas iteraciones están alejadas. Y el valor medio no concuerda con las amplitudes deseadas.

Otra manera de analizar este resultado, es a través de un Criterio de Información. En esta oportunidad, se utilizó AIC (Akaike Information Criteria). Su definición es:

$$\mathrm{AIC} = -2\log\mathcal{L} + 2k + \frac{2k(k+1)}{N-k-1}$$

En que $\mathcal{L}$ corresponde a la función de verosimilitud evaluada en los parámetros que lo maximizan (en este caso, la media del conjunto de los parámetros entregados por MCMC). k es el número total de parámetros que tiene el modelo y N es el número de datos utilizados para la estimación. Para este problema, $N = 100$.

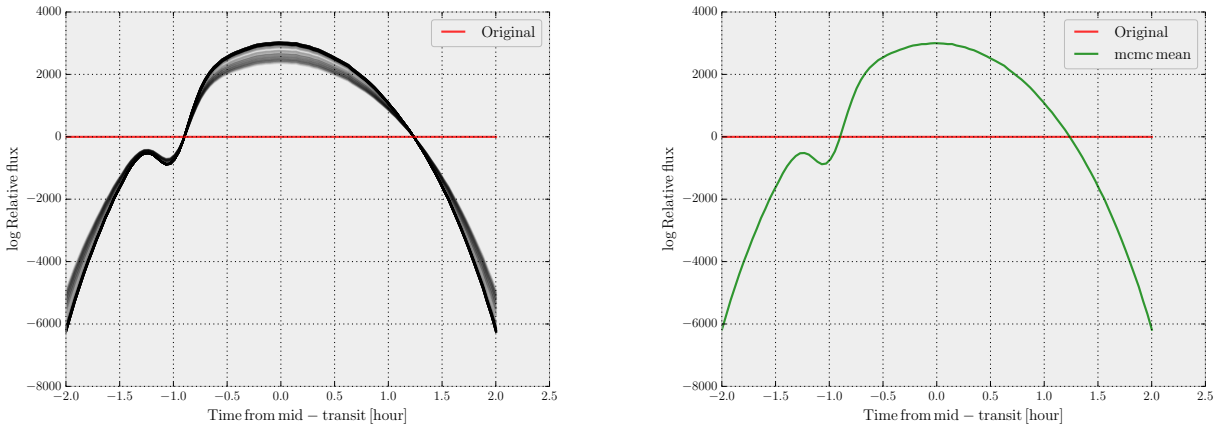

Figura 6: Resultado de MCMC para 9 componentes de PCA consideradas

Se calculó este indicador para las etimaciones con diferente número de componentes de pca incorporadas al modelo. De este modo, el rango de k va entre 6 y 14. Con esto, el modelo nunca es demasiado sencillo como para subestimar los datos. Gráficamente, esto puede verse en el panel de la izquiera de la Figura 7.

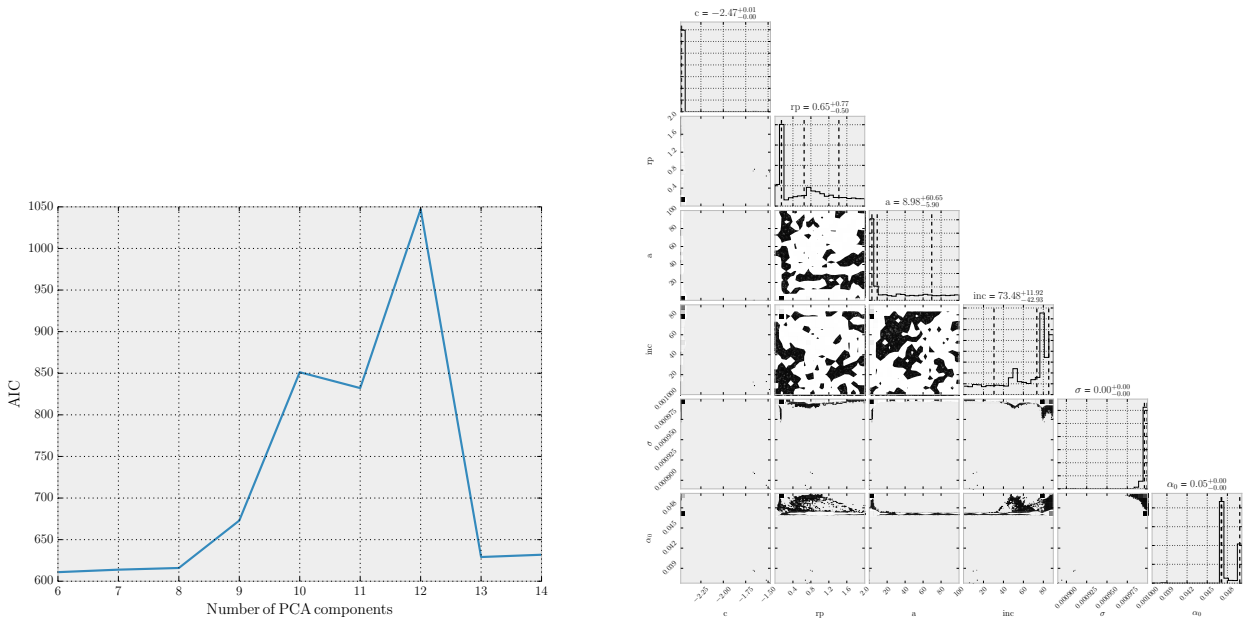

Figura 7: Resultados de MCMC. Izquierda: AIC calculado con diferente número de componentes. Derecha: Distribuciones a-posteriori para una componente incorporada

En el gráfico, puede verse que el valor mínimo se alcanza con el menor número de componentes, 1. Con cualquier número mayor, el modelo no ajusta tan bien los datos.

Este resultado concuerda con lo expresado en el cálculo de las componentes de PCA. En él, se determinó que la componente más importante es la primera. Y que si el modelo solo consideraba esta componente, no habría grandes pérdidas en la información contenida en el modelo.

Para comprender cómo trabaja MCMC, se han graficado las distribuciones a-posteriori de los parámetros estimados. Tomando en cuenta que, si se considera solo la primera componente de pca, la estimación es

la mejor, se muestra las distribuciones solo para esta configuración. El gráfico puede verse en el panel de la derecha en la Figura 7.

En él, puede notarse que las distribuciones se encuentran poco extendidas. También, se ve que los parámetros del modelo de tránsito tienen alta dispersión. Esto puede deberse a que el modelo utilizado, `batman`, tiene baja sensibilidad a la variación de estos tres parámetros utilizados.

c) Para intentar modelar de mejor manera el ruido asociado a los datos, se utilizan Procesos Gausianos. A través del módulo `George` se intenta asociar una estructura de correlación al ruido.

En esta ocasión, y utilizando las posibilidades del módulo ya mencionado, se asumirá que el ruido proviene de un Kernel Matérn-3/2.

Al utilizar Procesos Gaussianos en un problema de optimización (o ajuste de una curva), se agregan dos parámetros más a determinar con MCMC. Se incorpora la amplitud del kernel y su posición (localización).

Como en la sección anterior, se considera solo la primera componente de pca para este modelo, con lo que habrá que determinar valores para 8 parámetros.

Ejecutando MCMC con el ruido incorporado como un Proceso Gausiano, se obtienen los resultados mostrados en la Figura 8.

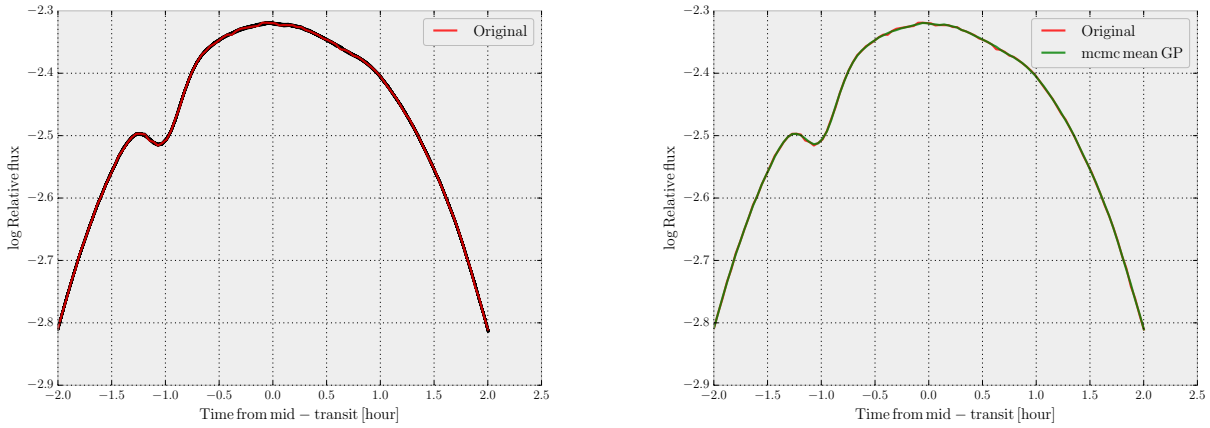

Figura 8: Resultados de MCMC con ruido proveniente de un Proceso Gaussiano. Izquierda: Datos originales y samples al azar de MCMC. Derecha: Datos originales y valor medio de las ejecuciones de MCMC

Sin demasiada dificultad, puede notarse que los valores entregados por MCMC se acercan mucho más a los valores originales. Se puede decir que incorporar un Proceso Gaussiano para el ruido de los datos (en vez de considerar ruido blanco) mejoró la estimación de parámetros del modelo completo.

Para verlo con un ejemplo concreto, se puede tomar la distribución marginal del parámetro rp del modelo de tránsito. En la Figura 9, se pueden ver las diferencias entre considerar, o no, un Proceso Gaussiano en el ruido.

A partir de estos gráficos, se puede ver que el parámetro pasa de tener una distribución concentrada en el borde del intervalo entregado por la distribución a-priori a mostrar una distribución centrada en un punto independiente de los bordes. Esto puede indicar, entre otras cosas, que incorporar un Proceso Gaussiano en el ruido, mejoró el comportamiento de los parámetros.

Ante esta evidencia (comparación del ajuste con datos originales y distribuciones marginales), se puede concluir que agregar un Proceso Gaussiano mejora considerablemente el comportamiento de MCMC y la estimación de parámetros en el modelo utilizado.

d) Asumir que los datos contienen ruido blanco es una suposición que, entre otras razones, está basada en cuánto se simplifican los cálculos y estimaciones de parámetros en modelos. Solamente en los últimos años,

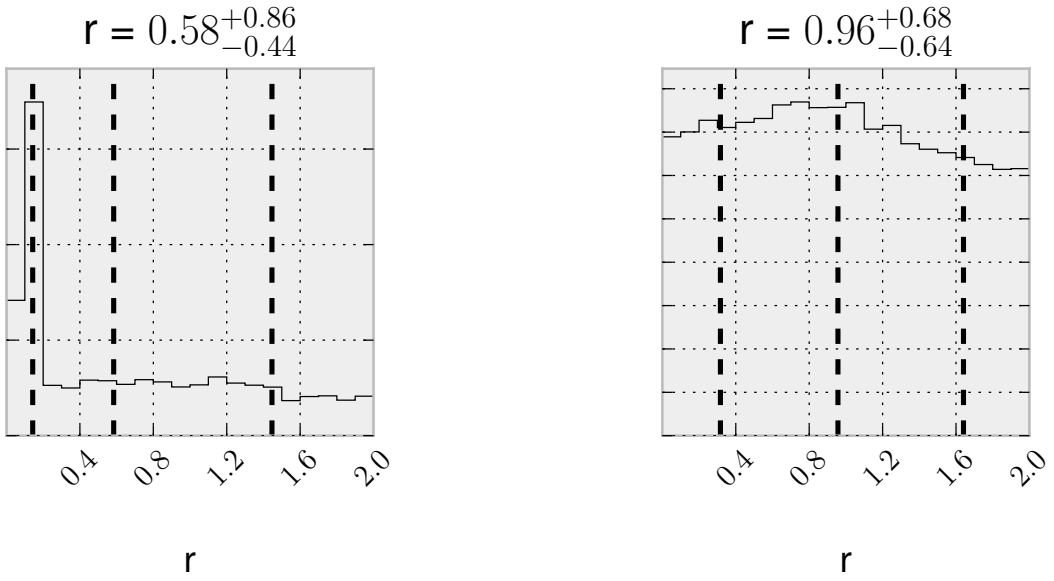

Figura 9: Distribución marginal a-posteriori para el parámetro rp del modelo de tránsito. Izquierda: considerando ruido blanco. Derecha: considerando un Proceso Gaussiano para el ruido

la computación ha permitido incorporar características más complejas en los modelos y en los métodos para resolverlos y estimar sus parámetros.

Es así que incorporar Procesos Gaussianos es un paso que se acerca más a la realidad de las mediciones (en Astronomía al menos). En principio, no se sabe la estructura que tiene el ruido de las mediciones. Y es mucho más probable que éste tenga algún tipo de correlación debida a los métodos de medición o a errores del observador y otros factores ambientales.

Por lo tanto, ante la opción de considerar ruido blanco o una estructura de correlación en el ruido, es preferible usar Procesos Gaussianos para mejorar la estimación de parámetros de un modelo dado.

Utilizando la información entregada por el paquete `corner` (parte de `emcee`) y que puede verse en la parte superior de cada panel de la Figura 9, se entrega una estimación para el parámetro rp. Esta estimación considera una banda de credibilidad de 68 % alrededor del parámetro. Esto corresponde a una estimación de $1\,\sigma$ del parámetro.

$$\hat{\mathrm{rp}}_{\mathrm{GP}} = 0{,}96^{+0{,}68}_{-0{,}64}$$

Se puede ver que la banda de credibilidad es bastante grande. Esto puede deberse, entre otras razones a la gran cantidad de parámetros que conforman el modelo estudiado.