Procesamiento y Análisis de Datos Astronómicos

8.- Modelado de datos y determinación de parámetros I

R. Gil-Hutton

Marzo 2020

Actividades:

Práctica 7:

- Si se extraen muestras de quasares del survey de Parkes (Masson & Wall 1977) seleccionándolos según si su espectro en radio es plano o no, se obtienen los siguientes valores:
  flat=0,1,0,0,1,0,1,8,7,11,6,7,5,10,6,5,3,0
  noflat=0,0,0,0,0,1,0,0,2,2,1,3,9,4,5,4,3,0
  para sus respectivos histogramas que corresponden a bines de 0.50 de ancho centrados en magnitudes que van desde 12.75 hasta 21.25. Si asumimos que H_0 es "ambas muestras tienen la misma distribución" contra H_1 "las muestras están desplazadas", decida que prueba de hipótesis no paramétrica utilizaría e indique a que conclusión llega.

Actividades:

Práctica 7 (cont.):

- Repita el ejercicio anterior utilizando una prueba no paramétrica diferente. Compare los resultados.
- Agrege a la muestra de quasares con espectro no plano 10 elementos más al azar utilizando un generador arbitrario y repita la prueba. Compare los resultados.
- Desplace la muestra de quasares con espectro no plano media magnitud hacia valores más bajos y repita la prueba. Comente sus conclusiones.

Actividades:

Práctica 7 (cont.):

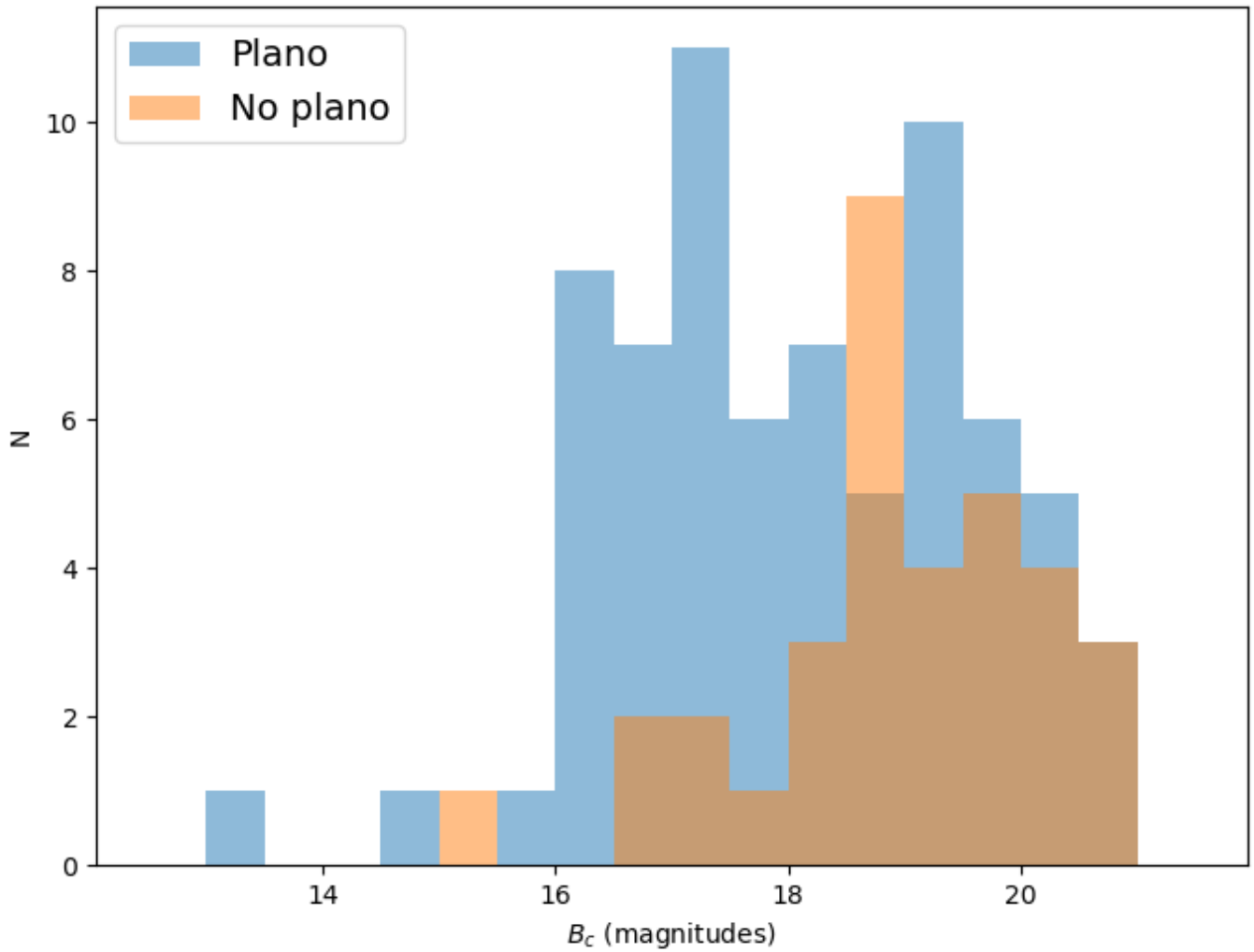
- Supongamos que adquirimos una muestra con los siguientes valores:
  [44, 23, 77, 38, 68, 41, 98, 5, 27, 74, 36, 33, 80, 27, 74, 39, 34, 33, 59, 55]
  y queremos comprobar la hipótesis H_0 "la mediana de la población de donde se obtuvo es de 30". Elija una prueba de hipótesis y comente los resultados.
- Utilizando la muestra del ejercicio anterior compruebe el grado de aleatoriedad respecto de un valor medio de 40 para H_0: "la muestra es aleatoria".

Actividades:

Práctica 7 (cont.):

- Utilizando una de las variables de su archivo de datos como población, tome dos muestras al azar de 100 elementos cada una y compruebe que pertenecen a la misma población.

Actividades:



Actividades:

Si se usa la prueba de Mann-Whitney hay que reconstruir la muestra porque inicialmente tenemos histogramas. Asumo siempre $\alpha = 0{,}05$:

```
In [166]: noflat=[0,0,0,0,0,1,0,0,2,2,1,3,9,4,5,4,3,0]

In [167]: flat=[0,1,0,0,1,0,1,8,7,11,6,7,5,10,6,5,3,0]

In [168]: bins=np.linspace(12.75,21.25,18)

In [169]: nff=[]
     ...: for ii in range(18):
     ...:    for jj in range(noflat[ii]):
     ...:       nff.append(bins[ii])
     ...:

In [170]: ff=[]
     ...: for ii in range(18):
     ...:    for jj in range(flat[ii]):
     ...:       ff.append(bins[ii])
     ...:

In [171]: sts.mannwhitneyu(ff,nff,alternative='less')
Out[171]: MannwhitneyuResult(statistic=809.5, pvalue=0.003133999900085506)

In [172]: sts.mannwhitneyu(ff,nff,alternative='greater')
Out[172]: MannwhitneyuResult(statistic=809.5, pvalue=0.9969309092979012)
```

Actividades:

Si comparamos los histogramas con la prueba de Wilcoxon encontramos que son diferentes y si comparamos las muestras con la prueba de Anderson-Darling para varias muestras nos indica que provienen de distribuciones diferentes:

```
In [182]: sts.wilcoxon(flat,noflat)
Out[182]: WilcoxonResult(statistic=11.0, pvalue=0.014737899259373008)

In [183]: sts.anderson_ksamp([ff,nff])
/usr/lib/python3/dist-packages/scipy/stats/morestats.py:1680: UserWarning: appro
ximate p-value will be computed by extrapolation
  warnings.warn("approximate p-value will be computed by extrapolation")
Out[183]: Anderson_ksampResult(statistic=4.690688138746392, critical_values=arra
y([0.325, 1.226, 1.961, 2.718, 3.752]), significance_level=0.004616732805469135)
```

Actividades:

Se agregan 10 elementos al azar a la muestra de objetos con espectro no plano:

```
In [185]: new=nff[:]

In [186]: ex=np.random.randint(0,18,10)

In [187]: ex
Out[187]: array([ 3,  7, 12,  5, 17,  8,  0,  0,  6, 14])

In [188]: for ii in range(len(ex)):
     ...:     new.append(bins[ex[ii]])
     ...:

In [189]: sts.mannwhitneyu(ff,new,alternative='less')
Out[189]: MannwhitneyuResult(statistic=1317.0, pvalue=0.07875738411856892)

In [190]: sts.mannwhitneyu(ff,new,alternative='greater')
Out[190]: MannwhitneyuResult(statistic=1317.0, pvalue=0.9220884814215349)
```

Actividades:

Se desplaza 0,5 magnitudes hacia valores más bajos a la muestra de objetos con espectro no plano:

```
In [207]: new=np.array(nff)-0.5

In [208]: sts.mannwhitneyu(ff,new,alternative='less')
Out[208]: MannwhitneyuResult(statistic=1029.0, pvalue=0.11069088229500479)

In [209]: sts.mannwhitneyu(ff,new,alternative='greater')
Out[209]: MannwhitneyuResult(statistic=1029.0, pvalue=0.8906049161221876)
```

Para comparar medianas se utiliza la prueba de Kruskal-Wallis. La segunda muestra es artificial con mediana de 30:

```
In [217]: muestra=[44, 23, 77, 38, 68, 41, 98, 5, 27, 74, 36, 33, 80, 27, 74,
     ...: 39, 34, 33, 59, 55]

In [218]: test=np.ones(len(muestra))*30.

In [219]: sts.kruskal(muestra,test)
Out[219]: KruskalResult(statistic=12.042457381794783, pvalue=0.0005200237101840678)
```

Actividades:

```
In [249]: ale=np.array(muestra)-40.

In [250]: ale
Out[250]:
array([  4., -17.,  37.,  -2.,  28.,   1.,  58., -35., -13.,  34.,  -4.,
        -7.,  40., -13.,  34.,  -1.,  -6.,  -7.,  19.,  15.])

In [251]: inx=np.where(ale > 0.)

In [252]: ma=len(ale[inx])

In [253]: inx=np.where(ale < 0.)

In [254]: md=len(ale[inx])

In [255]: r=13

In [256]: 2.*spc.factorial(ma-1)/spc.factorial(int(r/2-0.5))/spc.factorial(int(m
     ...: a-1-r/2+0.5))*spc.factorial(ma-1)/spc.factorial(int(r/2-1.5))/spc.fact
     ...: orial(int(ma-1-r/2+1.5))/(spc.factorial(2*ma)/spc.factorial(ma)/spc.fa
     ...: ctorial(ma))
Out[256]: 0.11457273376778021

In [257]: mu=2*ma*md/(ma+md)+1

In [258]: sig=np.sqrt(2.*ma*md*(2*ma*md-20)/20**2/19)

In [259]: sts.norm.ppf(0.975,loc=mu,scale=sig)
Out[259]: 15.265721965564385

In [260]: sts.norm.ppf(0.025,loc=mu,scale=sig)
Out[260]: 6.734278034435613
```

Modelado de datos:

El objetivo del modelado de datos es estimar los parámetros que caracterizan una población a partir de estimadores obtenidos de una muestra aleatoria. Para lograr el objetivo se debe:

1. Conocer o modelar la distribución de probabilidades de la población.
2. Disponer de una muestra aleatoria representativa de la población.
3. Aplicar un procedimiento par obtener estimadores de los parámetros de la población.

En general, no es posible cumplir con algunas de estas condiciones.

Distribución de probabilidades:

1. Hay varias opciones para obtener un modelo de la distribución de probabilidades de la población:
   - Cuando resulta posible se utilizan distribuciones conocidas o las que están disponibles en scipy.stats.
   - En el caso que lo anterior no sea posible se utilizan generalizaciones multiparamétricas de distribuciones conocidas.
   - En casos muy convenientes es posible utilizar distribuciones especiales, empíricas o no, basadas generalmente en información física.
   - Si las opciones anteriores no son posibles se puede obtener una distribución empírica mediante una función de interpolación construida mediante un polinomio, función racional, etc.

Distribución de probabilidades:

- La idea de utilizar generalizaciones multiparamétricas de distribuciones conocidas es emplear una distribución conocida y deformar su distribución.

a) El ejemplo más utilizado en este tipo de generalización es la serie de Gram-Charlier:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z^2}{2}\right) \times \left(1 + \sum_i a_i H_i(z)\right)$$

para $z = (x - \mu)/\sigma$, donde los a_i son los parámetros a determinar y las H son polinomios de Hermite:

$$H_n(x) = (-1)^n e^{x^2/2} \frac{d^n}{dx^n} e^{-x^2/2}$$

Distribución de probabilidades:

b) Los polinomios de Hermite son ortogonales para una función de peso $\exp(-z^2/2)$, por lo que sus coeficientes están relacionados con los momentos de la distribución que se crea.

c) Los polinomios de Hermite de orden par producen un cambio de escala en la distribución a distorsionar, mientras que los de orden impar modifican tanto escala como posición.

d) Para una distribución basada en una exponencial $\exp(-x/a)$ la función de distorsión está formada por polinomios de Laguerre.

e) Para una distribución basada en una función del tipo $x^{\alpha}(1-x)^{\beta}$ la función de distorsión está formada por polinomios de Jacobi.

Muestra representativa:

2. Cuándo una muestra aleatoria es representativa de la población?. Cuál es el número mínimo de elementos necesario?.

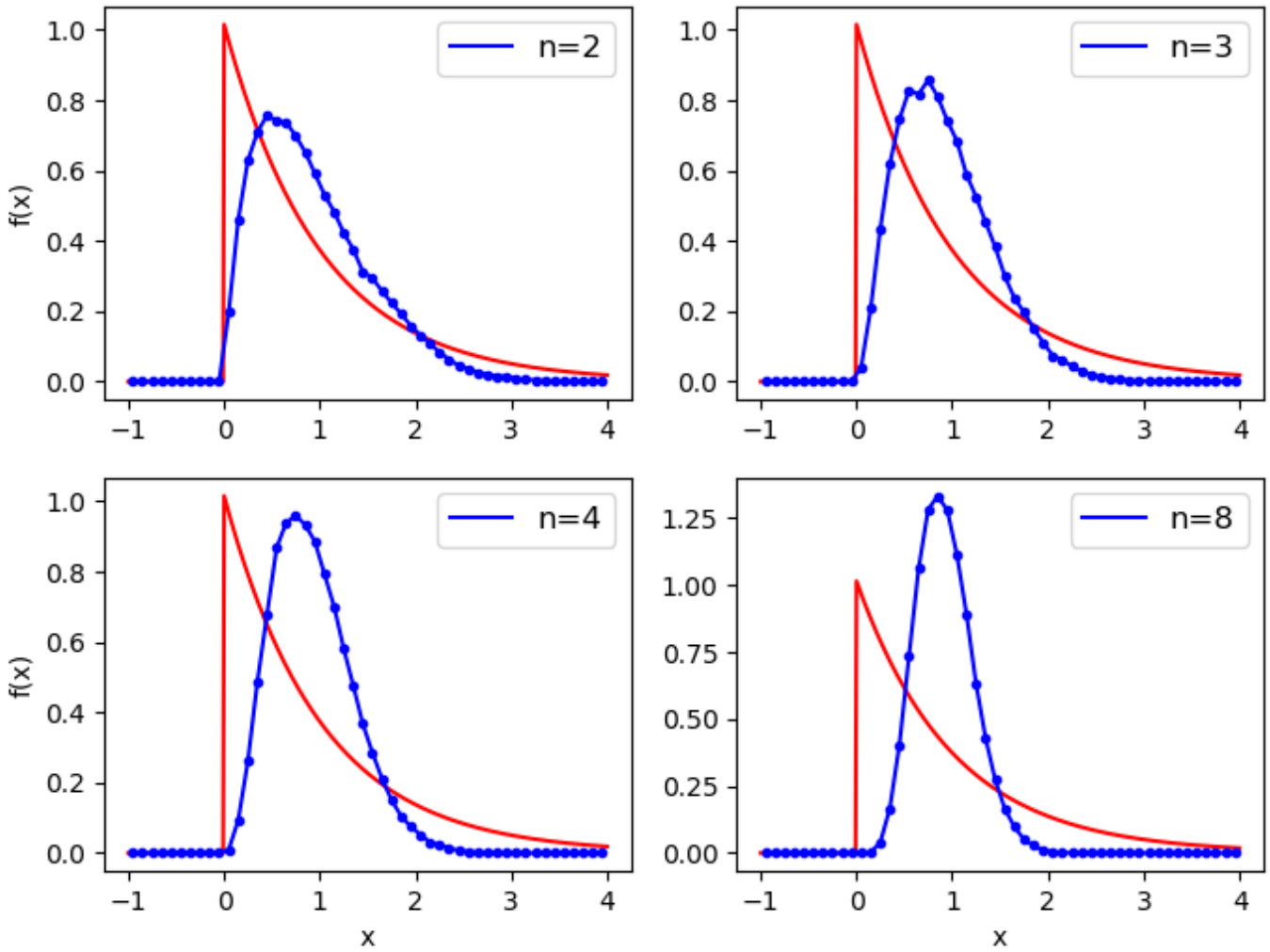
a) No es una pregunta sencilla de contestar y depende fuertemente de las características de la población a estudiar.

b) Lo mejor es tener en cuenta los errores esperados en los estimadores. El Teorema Central de límite nos dice que:

$$\bar{Z}_k = \frac{\bar{x_k} - \mu}{\sigma/\sqrt{k}} \quad \rightarrow \quad s_{\bar{x}} = \frac{\sigma}{\sqrt{k}}$$

d) Entonces, el error relativo $s_{\bar{x}}/\sigma \sim 1/\sqrt{k}$ y es posible fijar k en función del error relativo de $\bar{x}$.

Muestra representativa:



Estimadores:

3. La forma preferida para obtener estimadores de los parámetros de la población es el método de máxima verosimilitud (Maximum-likelihood en inglés):

a) Consideremos la distribución de probabilidades $f(x, \mathbf{a})$, donde x es una variable aleatoria y $\mathbf{a}$ es un vector de paŕametros que se quiere estimar.

b) Si $x_1, x_2, \cdots, x_k$ son datos independientes obtenidos de $f(x|\mathbf{a})$, la función de verosimilitud es:

$$\begin{aligned} \mathcal{L}(x_1, x_2, \cdots, x_k) &= f(x_1, x_2, \cdots, x_k, \mathbf{a}) = \\ &= f(x_1|\mathbf{a}) f(x_2|\mathbf{a}) \cdots f(x_k|\mathbf{a}) = \\ &= \prod_{i=1}^{k} f(x_i|\mathbf{a}) \end{aligned}$$

Estimadores:

c) La idea básica del método es elegir la aproximación para **a** para la cual el valor de $\mathcal{L}$ sea tan grande como sea posible.

d) Si $\mathcal{L}$ es una función derivable de **a**, una condición necesaria para que $\mathcal{L}$ tenga un máximo es:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{a}} = 0$$

e) Como $f(x|\mathbf{a})$ es no negativa y $\ln \mathcal{L}$ es monótona creciente y tiene un máximo precisamente donde los tiene $\mathcal{L}$, podemos escribir:

$$\frac{\partial \ln \mathcal{L}}{\partial \mathbf{a}} = 0$$

que representa una ventaja al reemplazar productos por sumas.

Estimadores:

f) La ecuación resultante no siempre es fácil de resolver, pero si para un parámetro existe un estimador eficiente siempre es posible encontrarlo a partir de ella.

g) Hay que recordar que el estimador de máxima verosimilitud encontrado, $\hat{\mathbf{a}}$, es un estadístico más (depende solo de los datos) y que puede ser insesgado:

Ejemplo: Si aplicamos el método de máxima verosimilitud a una variable aleatoria x con una distribución normal, media μ y varianza σ^2, tenemos que su distribución de probabilidades es:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Estimadores:

Su función de verosimilitud es:

$$\mathcal{L}(x|\mu;\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x_1-\mu)^2}{2\sigma^2}\right]\cdots\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(x_k-\mu)^2}{2\sigma^2}\right]$$

$$\mathcal{L}(x|\mu;\sigma) = \left(\frac{1}{\sqrt{2\pi}}\right)^k\left(\frac{1}{\sigma}\right)^k\exp\left[-\frac{\sum_{i=1}^{k}(x_i-\mu)^2}{2\sigma^2}\right]$$

tomando logaritmos:

$$\ln\mathcal{L}(x|\mu;\sigma) = -k\ln\left(\sqrt{2\pi}\right) - k\ln\sigma - \frac{\sum_{i=1}^{k}(x_i-\mu)^2}{2\sigma^2}$$

Estimadores:

La primera ecuación de condición es:

$$\frac{\partial \ln \mathcal{L}}{\partial \mu} = \frac{\sum_{i=1}^{k}(x_i - \mu)}{\sigma^2} = 0$$

y la segunda:

$$\frac{\partial \ln \mathcal{L}}{\partial \sigma} = -\frac{k}{\sigma} + \frac{\sum_{i=1}^{k}(x_i - \mu)^2}{\sigma^3} = 0$$

Estimadores:

De la primera se obtiene que:

$$\sum_{i=1}^{k}(x_i - \mu) = \sum_{i=1}^{k} x_i - k\mu = 0 \quad \rightarrow \quad \hat{\mu} = \bar{x} = \frac{\sum_{i=1}^{k} x_i}{k}$$

y de la segunda:

$$-\frac{k}{\sigma} + \frac{\sum_{i=1}^{k}(x_i - \mu)^2}{\sigma^3} = 0 \quad \rightarrow \quad \hat{\sigma^2} = s^2 = \frac{\sum_{i=1}^{k}(x_i - \bar{x})^2}{k}$$

Hay que notar que a partir del análisis del método de máxima verosimilitud se obtiene el método de mínimos cuadrados ordinario.

Estimadores:

h) Si resulta muy difícil obtener estimadores mediante el método de máxima verosimilitud, la opción usada frecuentemente es obtener los estimadores a partir de un ajuste numérico.
Si este es el caso, para un sistema de ecuaciones lineal se prefiere resolverlo mediante descomposición en valores singulares (Singular value decomposition (SVD) en inglés). En un caso no lineal el método preferido para resolver las ecuaciones es el de Levenberg-Marquardt. Ambos métodos están disponibles en **Python** como funciones en Numpy y Scipy (numpy.linalg.svd y scipy.optimize.least_squares, usando el método "lm").

Estimadores (SVD):

Una ecuación lineal de la forma:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z^2}{2}\right) \times \left(1 + \sum_i a_i H_i(z)\right)$$

puede escribirse como $\mathbf{M} * \mathbf{a} = \mathbf{p}$, donde $\mathbf{M}$ es la matriz:

$$\mathbf{M} = \begin{pmatrix} fac_1 * H_1(z_1) & fac_1 * H_2(z_1) & \cdots & fac_1 * H_i(z_1) \\ fac_2 * H_1(z_2) & fac_2 * H_2(z_2) & \cdots & fac_2 * H_i(z_2) \\ \vdots & \vdots & \vdots & \vdots \\ fac_n * H_1(z_n) & fac_n * H_2(z_n) & \cdots & fac_n * H_i(z_n) \end{pmatrix}$$

donde $fac_j = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z_j^2}{2}\right)$,

Estimadores (SVD):

y tenemos a **a** y **p** que son los vectores:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_i \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} p(z_1) - fac_1 \\ p(z_2) - fac_2 \\ \vdots \\ p(z_n) - fac_n \end{pmatrix}$$

La idea es calcular la inversa de **M** por SVD para encontrar $\mathbf{a} = \mathbf{M}^{-1} * \mathbf{p}$.

Estimadores (SVD):

Ejemplo: Vamos a tratar de ajustar una distribución normal generalizada a la muestra de inclinaciones de objetos Apollo del archivo **apollo-aeih.dat**, utilizando un modelo lineal. La muestra cuenta con 9239 elementos lo que me permite una precisión relativa en los estimadores del 1 %.

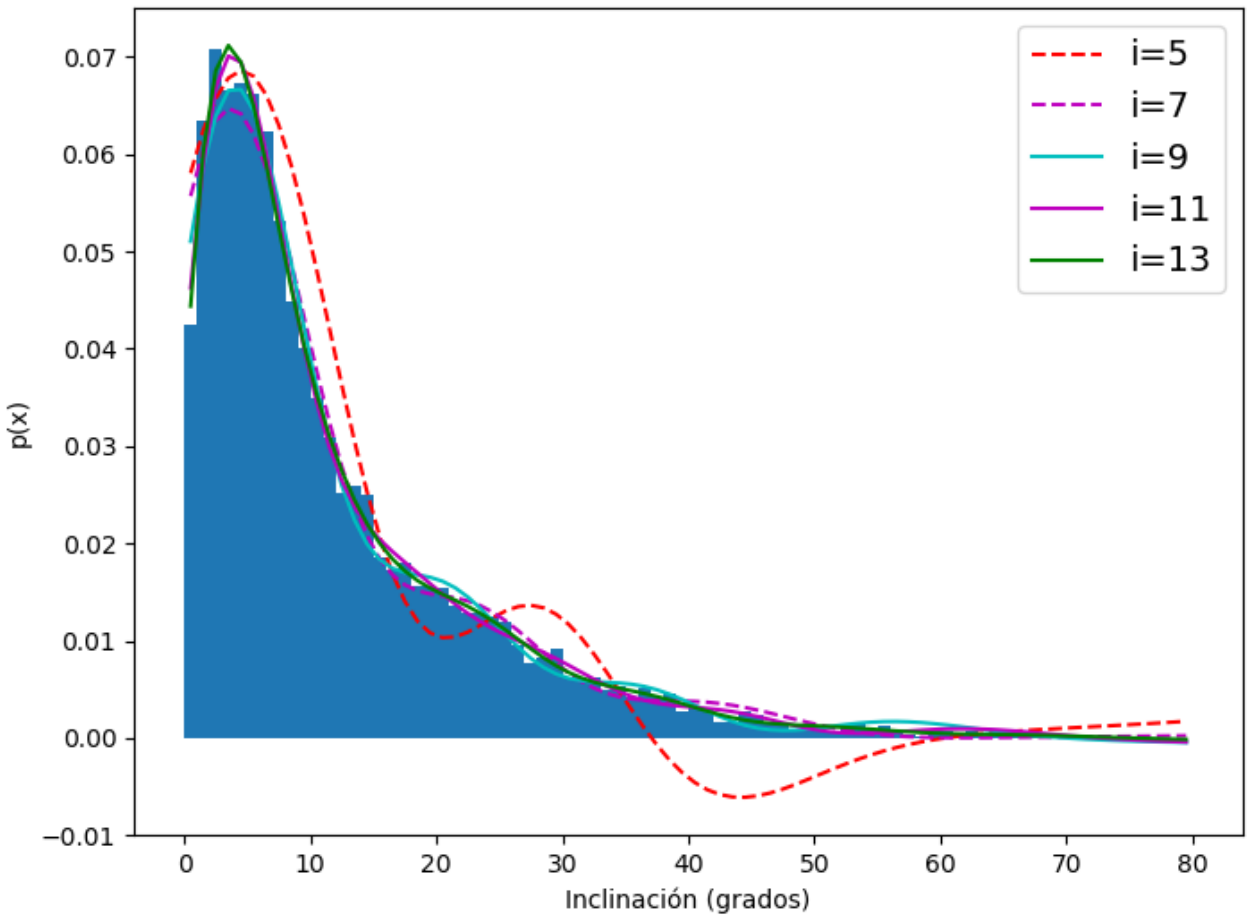
```
In [2]: apo=np.loadtxt('apollo-aeih.dat')

In [3]: vmed=np.mean(apo[:,2])

In [4]: vstd=np.std(apo[:,2],ddof=1)

In [5]: his,lim=np.histogram(apo[:,2],bins=80,range=(0,80))

In [6]: his=his/np.sum(his)

In [7]: xx=lim[1:]-0.5

In [8]: zz=(xx-vmed)/vstd

In [9]: fac=1./np.sqrt(2*np.pi)/vstd*np.e**(-zz**2/2)

In [10]: pp=his-fac
```

Estimadores (SVD):

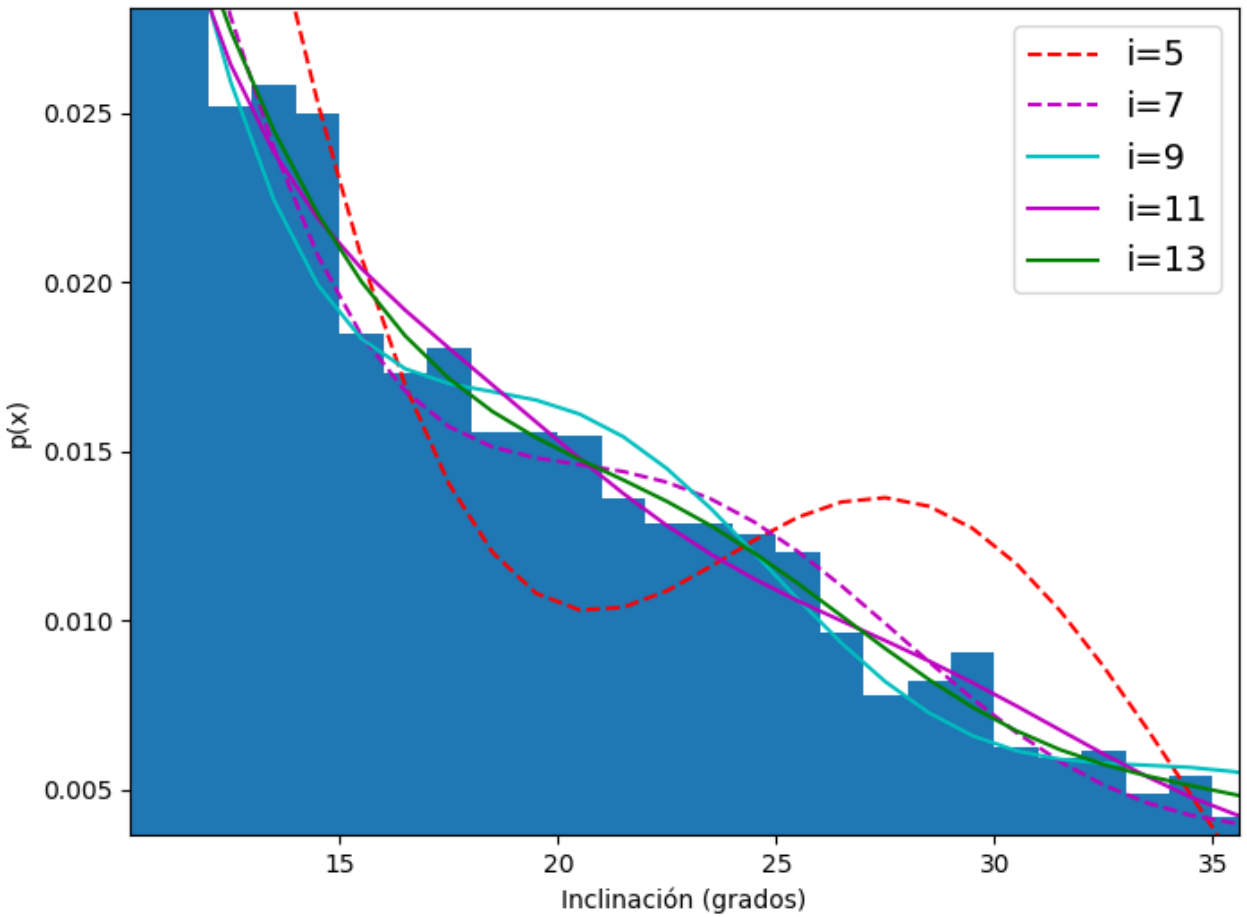
El procedimiento para hacer SVD está muy bien explicado en "Numerical Recipes" de Press et al.:

```
In [12]: import numpy.linalg as lin

In [13]: import scipy.special as spc

In [14]: mm=np.vstack((fac*spc.eval_hermitenorm(1,zz), fac*spc.eval_hermitenorm(
    ...: 2,zz), spc.eval_hermitenorm(3,zz),fac*spc.eval_hermitenorm(4,zz), fac*s
    ...: pc.eval_hermitenorm(5,zz))).T

In [15]: uu,ss,vv=lin.svd(mm,full_matrices=False)

In [16]: ss=np.diag(ss**(-1))

In [17]: ww=np.dot(ss,uu.T)

In [18]: mminv=np.dot(vv.T,ww)

In [19]: aa=np.dot(mminv,pp)

In [20]: res=np.dot(mm,aa)+fac

In [21]: res=res/np.sum(res)
```

Estimadores (SVD):



Estimadores (SVD):



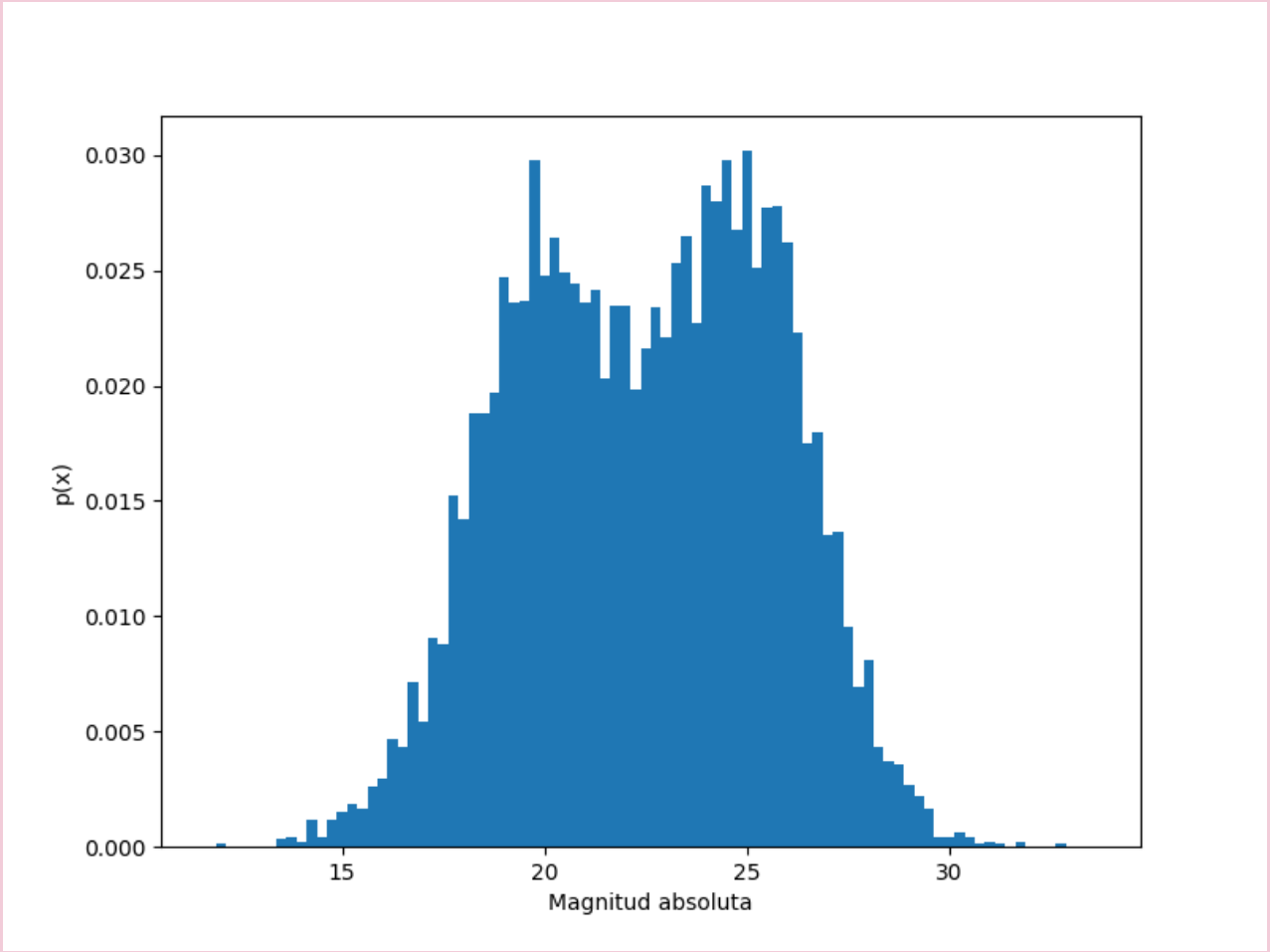
Estimadores (L-M):

Ejemplo: Vamos a tratar de ajustar una distribución bimodal usando una combinación de distribuciones normales a la muestra de magnitudes absolutas de objetos Apollo del archivo **apollo-aeih.dat**, utilizando un modelo no lineal. La muestra cuenta con 9239 elementos lo que me permite una precisión relativa en los estimadores del 1 %.

```
In [97]: apo=np.loadtxt('apollo-aeih.dat')

In [98]: np.min(apo[:,3]),np.max(apo[:,3])
Out[98]: (12.4, 33.24)

In [99]: his,lim=np.histogram(apo[:,3],bins=88,range=(12,34))

In [100]: his=his/np.sum(his)

In [101]: xx=lim[1:]-0.125
```

Estimadores (L-M):

La distribución de magnitudes para los objetos Apollo es la siguiente:



Estimadores (L-M):

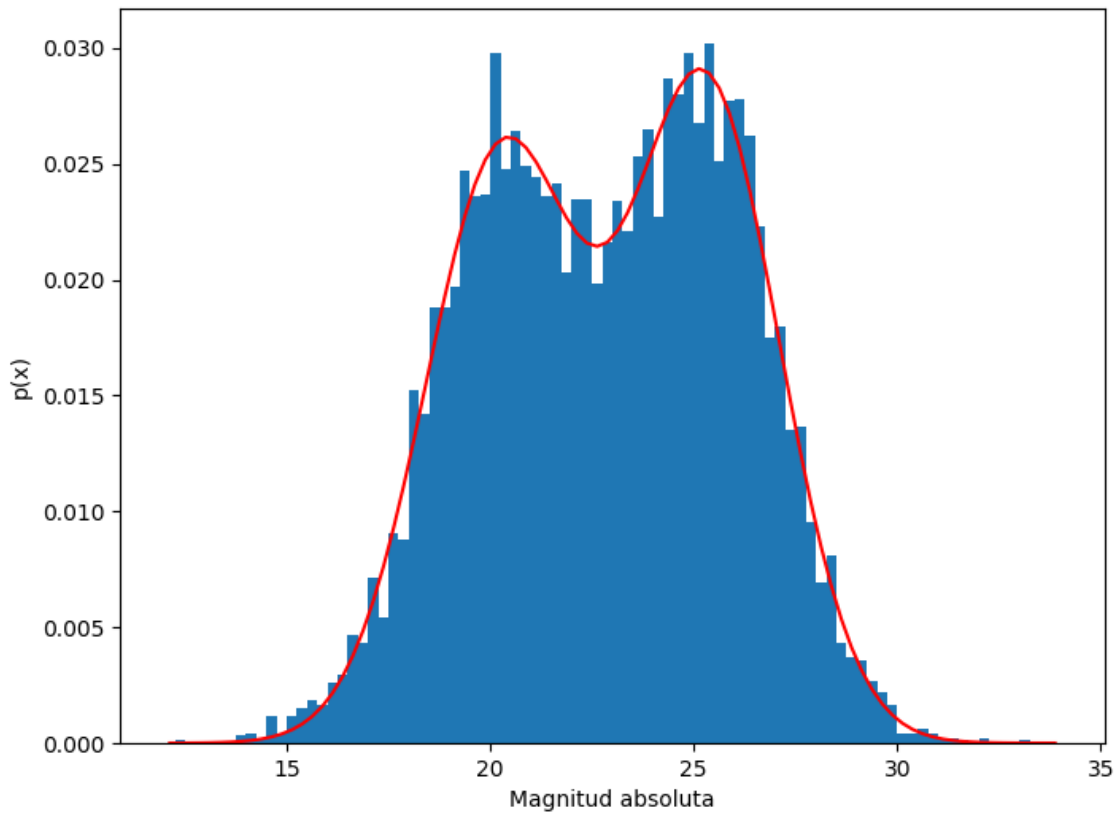
La función (arbitraria) usada para el ajuste es:

$$p(x) = \frac{k_1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right) + \frac{k_2}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)$$

```
In [102]: def funcajus(par,xx,yy):
     ...:     """
     ...:     El array par contiene mu1,var1,mu2,var2,k1 y k2.
     ...:     Los arrays xx e yy tienen los valores centrales de los bines y su
valor en la distribucion, respectivamente.
     ...:
     ...:     La función devuelve las diferencias entre el ajuste y el valor obs
ervado en la distribución para cada bin.
     ...:     """
     ...:     zz=par[4]/np.sqrt(2.*np.pi)/np.sqrt(par[1])*np.e**(-(xx-par[0])**2
/2/par[1])+par[5]/np.sqrt(2.*np.pi)/np.sqrt(par[3])*np.e**(-(xx-par[2]
)**2/2/par[3])-yy
     ...:     return zz
     ...:

In [103]: import scipy.optimize as opt

In [104]: sol=opt.least_squares(funcajus,np.array([20,2,25,2,1.,1.]),method='lm'
     ...: ,ftol=1.e-12, gtol=1.e-12,xtol=1.e-12,args=(xx,his))
```

Estimadores (L-M):

```
In [109]: sol['x']
Out[109]:
array([20.28165865,  3.5040597 , 25.28577549,  3.3455703 ,  0.11921901,
       0.12987199])
```



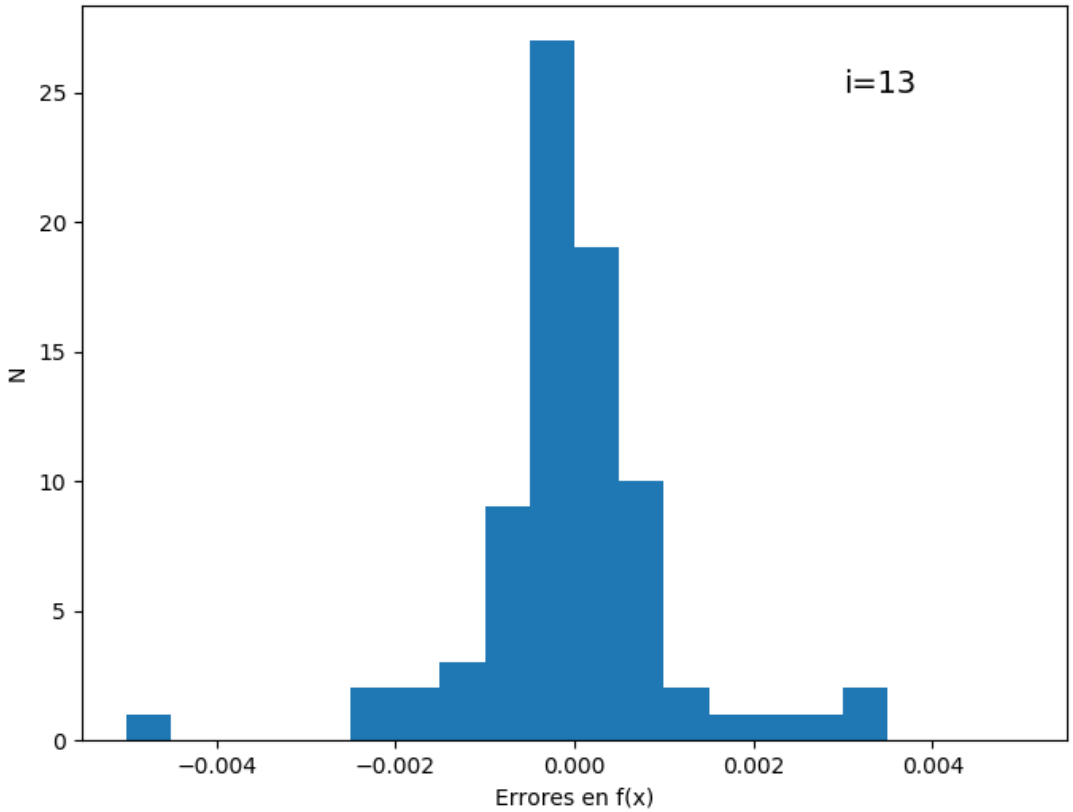
Errores en el modelo:

Un punto que siempre hay que tener en cuenta es que, a pesar de obtener un ajuste razonable, muchas veces el modelo elegido puede ser erróneo. Por ejemplo:

- Los errores del modelo no tienen una distribución normal o parecida a normal.
- Los errores del modelo no pasan una prueba de aleatoriedad.
- El estimador χ^2 no es aceptado por su prueba de hipótesis.
- Es posible que se estén considerando datos anómalos que afecten el ajuste y se requieran criterios más robustos.
- Tal vez exista un error en el ajuste de los parámetros y/o sea necesario incrementar su número.

Errores en el modelo:

Errores para el ajuste a la distribución de inclinación con una serie de Gram-Charlier de 13 parámetros:



Errores en el modelo:

Prueba de aleatoriedad de errores para el ajuste a la distribución de inclinación con una serie de Gram-Charlier de 13 parámetros:

```
In [292]: res=np.dot(mm,aa)+fac

In [293]: res=res/np.sum(res)

In [294]: inx=np.where(dif > 0.)

In [295]: ma=len(dif[inx])

In [296]: inx=np.where(dif < 0.)

In [297]: md=len(dif[inx])

In [298]: r=36

In [299]: mu=2.*ma*md/(ma+md)+1

In [300]: sig=np.sqrt(2.*ma*md*(2*ma*md-80)/80**2/79)

In [301]: vs=sts.norm.ppf(0.975)

In [302]: vi=sts.norm.ppf(0.025)

In [303]: vs*sig/np.sqrt(80)+mu
Out[303]: 116.74789255963168

In [304]: vi*sig/np.sqrt(80)+mu
Out[304]: -35.547892559631684

In [305]: mu
Out[305]: 40.6
```

Errores en el modelo:

Una forma de estimar el error en el ajuste de los N datos y_i con el modelo Y es calcular el siguiente estimador de máxima verosimilitud:

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - Y(x_i; \mathbf{a})}{\sigma_i} \right)^2$$

donde $\mathbf{a}$ es un vector de M parámetros del modelo y σ_i los errores para cada valor. Para estimar la probabilidad de obtener un cierto valor de χ^2 para $\nu = N - M$ grados de libertad se utilizan las funciones gamma incompletas: $\mathcal{P}(\chi^2|\nu)$ y $\mathcal{Q}(\chi^2|\nu) = 1 - \mathcal{P}(\chi^2|\nu)$ (scipy.special.gammainc() y scipy.special.gammaincc()).

Errores en el modelo:

- $\mathcal{P}(\chi^2|\nu)$ se define como la probabilidad de que el valor de χ^2 observado para un modelo correcto sea menor que un valor de referencia para un cierto número de grados de libertad. $\mathcal{P}(\chi^2|\nu)$ se calcula en **Python** con la función scipy.special.gammainc($\frac{\nu}{2}, \frac{\chi^2}{2}$).
- $\mathcal{Q}(\chi^2|\nu)$ se define como la probabilidad de que el valor de χ^2 obtenido exceda el valor de referencia para ciertos grados de libertad, aún si el modelo es correcto. $\mathcal{Q}(\chi^2|\nu)$ se calcula en **Python** con scipy.special.gammaincc($\frac{\nu}{2}, \frac{\chi^2}{2}$).

```
In [315]: chi2=np.sum(his-res)**2/80./0.0005**2

In [316]: nu=80-13

In [317]: spc.gammainc(nu/2.,chi2/2.)
Out[317]: 0.0

In [318]: spc.gammaincc(nu/2.,chi2/2.)
Out[318]: 1.0
```

Errores en el modelo:

Para estimar parámetros hay algunos métodos más robustos. Con un estimador de tipo M (estimadores de máxima verosimilitud) vimos antes que la función de verosimilitud es función de $f(x_i|\mathbf{a}) = \exp[-\rho(y_i; Y(x_i, \mathbf{a}))]$ donde $\rho(z)$ es función de $z = [y_i - Y(x_i, \mathbf{a})]/\sigma_i$.
Si definimos $\phi(z) = d\rho(z)/dz$, las ecuaciones de condición para cada parámetro del modelo tienen la forma:

$$\sum_i \frac{1}{\sigma_i}\phi(z)\frac{\partial Y(x_i, \mathbf{a})}{\partial a_j} = 0$$

Si asumimos que los errores del ajuste tienen una distribución normal, tenemos que:

$$\rho(z) = \frac{1}{2}z^2 \quad \phi(z) = z \quad p(x_i|\mathbf{a}) \sim exp\left(-\frac{[y_i - Y(x_i, \mathbf{a})]^2}{2\sigma_i^2}\right)$$

Errores en el modelo:

Si los errores se distribuyen según una doble exponencial tenemos que:

$$\rho(z) = |z| \quad \phi(z) = sgn(z) \quad p(x_i|\mathbf{a}) \sim exp\left(-\left|\frac{y_i - Y(x_i, \mathbf{a})}{\sigma_i}\right|\right)$$

con colas mucho más largas que para una distribución normal. Una distribución con colas aún más largas es la distribución de Lorentz donde:

$$\rho(z) = \log\left(1 + \frac{1}{2}z^2\right) \quad \phi(z) = \frac{z}{1 + \frac{1}{2}z^2}$$

$$p(x_i|\mathbf{a}) \sim \frac{1}{1 + \frac{1}{2}\left[\frac{y_i - Y(x_i,\mathbf{a})}{\sigma_i}\right]^2}$$

Actividades:

Práctica 8:

- Para dos de las variables de su archivo de datos estime la precisión en los parámetros para el tamaño de las muestras, ajuste modelos creados con generalizaciones multiparamétricas de su elección, grafique los errores y estime con una prueba de hipótesis si son aleatorios o no.

Entrega

Para la próxima clase

Por consultas: ricardo.gil-hutton@conicet.gov.ar
Grupo de Ciencias Planetarias - CUIM 2