Procesamiento y Análisis de Datos Astronómicos

5.- Descripción Estadística de Datos

R. Gil-Hutton

Marzo 2020

Actividades:

Práctica 4:

- Ordenar el archivo de datos con el que están trabajando fila por fila de menor a mayor según los elementos de una columna cualquiera a su elección y utilizando funciones de Numpy.
- Modificar la función de ordenamiento por selección que se mostró en esta clase para que pueda operar sobre un array con varias columnas ordenando fila por fila de menor a mayor según los elementos de una columna cualquiera.

Actividades:

Práctica 4 (cont.):

- Proponer un procedimiento para verificar que los ordenamientos realizados son correctos y aplicarlo a los dos casos anteriores.
- El archivo **landolt.dat** (va adjunto) contiene posiciones y valores fotométricos para las estrellas standard de Landolt. Ordenar este archivo según el valor de la magnitud R y guardarlo en un nuevo archivo.

Actividades:

```
In [140]: apo=np.loadtxt('apollo-aeih.dat')

In [141]: ind=np.argsort(apo[:,0],kind='quicksort')

In [142]: arr=apo[ind,:]

In [143]: test=arr[:-1,0]-arr[1:,0]

In [144]: np.where(test > 0)
Out[144]: (array([], dtype=int64),)

In [145]:
```

```
In [145]: arr[0,:]
Out[145]: array([ 1.0000555 ,  0.12475442,  6.390889  , 25.65      ])

In [146]: ind=np.where(apo[:,0] == arr[0,0])

In [147]: apo[ind,:]
Out[147]: array([[[ 1.0000555 ,  0.12475442,  6.390889  , 25.65      ]]])
```

Actividades:

```
def selec_array(arr,col):
  """
  Ordena un array por el metodo de seleccion
  segun una columna a eleccion
  """
  nn=len(arr[:,0])
  for ii in range(nn):
    menor=ii
    for jj in range(ii+1,nn):
      if(arr[jj,col] < arr[menor,col]):
        menor=jj

    arr[ii,:],arr[menor,:]=arr[menor,:],arr[ii,:]

  return
```

Actividades:

```
In [149]: %run ordena-array.py

In [150]: arr=np.copy(apo)

In [151]: selec_array(arr,0)

In [152]: arr[0,:]
Out[152]: array([ 1.0000555 ,  0.12475442,  6.390889  , 25.65      ])

In [153]:
```

Actividades:

```
In [68]: f=open('landolt.dat','r')

In [69]: ll=f.readlines()

In [70]: f.close()

In [71]: stars=np.array(ll[6:])

In [72]: rmag=np.zeros(len(stars))

In [73]: for ii in range(len(stars)):
    ...:     ss=stars[ii].rstrip('\n').split('\t')
    ...:     rmag[ii]=float(ss[4])-float(ss[7])
    ...:

In [74]: ind=np.argsort(rmag,kind='quicksort')

In [75]: orden=stars[ind]

In [76]: f=open('new-landolt.dat','w')

In [77]: f.writelines(ll[:6])

In [78]: f.writelines(orden)

In [79]: f.close()

In [80]:
```

Muestras:

- Población: es el conjunto de todos los valores posibles de una variable de la cual se quiere obtener información.
- Muestra: es un subconjunto de la población de una variable de la cual se quiere obtener información.
- Muestra aleatoria: es una muestra de una variable de la cual se quiere obtener información y que fue adquirida de manera aleatoria (elemento con igual chance de ser elegido).
- Muestra simple: o muestra de conveniencia, es una muestra de una variable de la cual se quiere obtener información y que fue adquirida de manera no aleatoria.

Hay otros tipos de muestras como, por ejemplo, la muestra pesada y la muestra agrupada.

Representación de muestras:

Si contamos con una muestra de una cierta población la podemos representar de diferentes maneras:

- El número de veces que aparece un valor en la muestra se llama frecuencia.
- La división de la frecuencia en el tamaño de la muestra se denomina frecuencia relativa y puede tomar valores en el rango $[0, 1]$. La suma de las frecuencias relativas de una muestra es siempre igual a 1.
- La suma de las frecuencias relativas de todos los valores menores o iguales a un valor dado es la frecuencia relativa acumulada.

Representación de muestras:

- La función que representa las frecuencias relativas de una muestra se llama distribución de frecuencia.
- La función que representa las frecuencias relativas acumuladas de una muestra se denomina distribución acumulada de frecuencia.
- Para representar estas distribuciones usualmente es mejor utilizar un histograma que agrupa los valores de la muestra dentro de ciertos rangos de la variable denominados intervalos de clases o bins.
- Cuando cada bin se representa con un rectángulo cuya área es igual a la frecuencia relativa correspondiente el área total del histograma es 1 y tenemos una distribución de densidad.

Representación de muestras:

Para evitar complicaciones innecesarias en la representación de una muestra agrupada en un histograma es necesario que:

- todos los bins tengan preferentemente el mismo ancho.
- se debe fijar un criterio para asignar valores que queden en los límites de dos bins sucesivos.
- se debe elegir cuidadosamente el número total de bins a utilizar.
- si se desea comparar muestras agrupadas es necesario utilizar el mismo agrupamiento en ambas.

Siempre se debe recordar que una muestra agrupada implica cierta pérdida de información.

Representación de muestras:

Como elegir el número de bines en un histograma:

- el criterio más simple es: $n_{bins} = int(N^{1/2} + 0{,}5)$.
- el criterio de Sturges es: $n_{bins} = int[\log_2(N) + 0{,}5] + 1$.
- el criterio de Rice es: $n_{bins} = int[2 \times N^{1/3} + 0{,}5]$.
- el criterio de Scott es:
  $n_{bins} = int[(max - min) \times N^{1/3}/(3{,}5 \times std) + 0{,}5]$.
- ...también se pueden elegir según la conveniencia particular!.

Estimadores de poblaciones y muestras:

- El valor medio, varianza y desviación standard de una muestra se indican como $\bar{x}$, s^2 y s, respectivamente.
- El valor medio, varianza y desviación standard de una población se indican como μ, σ^2 y σ, respectivamente.
- Los estadísticos, como $\bar{x}$ y s^2, son propiedades de los datos que los sumarizan, reducen o describen.
- Valores como μ o σ describen las distribuciones y no son estadísticos, son parámetros.
- Los datos pueden seguir una cierta distribución, conocida o no, y es posible utilizar los estadísticos de la muestra para estimar los parámetros de esa distribución.
- Esto último se consigue obteniendo los valores de esperanza o estimadores para cada parámetro de interés.

Estimadores de poblaciones y muestras:

- Formalmente, un estimador es la suma de todos los posibles valores de una función $g(x)$ pesados con el correspondiente valor dado en su distribución de frecuencia $f(x)$ (o su probabilidad de ocurrencia). En el caso de distribuciones contínuas:

$$E[g(x)] = \int g(x)f(x)dx$$

- Es posible pensar a los estimadores como el resultado a obtener luego de repetir un experimento muchas veces y promediar sus resultados.

Estimadores de poblaciones y muestras:

A partir de una muestra, frecuentemente pequeña, se pretende conocer cuál es su distribución real de frecuencias y se quiere hacer esto en forma eficiente y robusta, pero:

- El tamaño de la muestra afecta la determinación de estadísticos y de los parámetros de la distribución.
- Los estadísticos deben ser insesgados. El estimador debe representar bien el valor real del parámetro.
- Los estadísticos deben ser consistentes. El estimador obtenido de una muestra grande debe dar el valor correcto.
- Los estadísticos deben mostrar desviaciones mínimas respecto de los valores correctos.
- Los estadísticos deben ser robustos ante la aparición de datos anómalos.

Estimadores de poblaciones y muestras:

A partir de los elementos de una muestra extraída de una población:

- Media de una población:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

- Media de una muestra:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

$\bar{x}$ es un estimador insesgado de μ

Estimadores de poblaciones y muestras:

A partir de los elementos de una muestra extraída de una población:

- Varianza de una población:

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

- Varianza de una muestra:

$$s^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

hay un factor $n/(n-1)$ para para convertir a s^2 en un estimador insesgado de σ^2

Estimadores de poblaciones y muestras:

- Mediana de una nuestra es el valor que tiene un percentil de 50 %.
- Modo de una muestra es el valor más frecuente o, en un histograma, el valor donde se produce el máximo.
- Desviación media de una muestra:

$$\bar{\Delta x} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \bar{x}|$$

Estimadores de poblaciones y muestras:

- Sesgo u oblicuidad de una muestra mide la pérdida de simetría de su distribución respecto del valor medio (aparición de colas):

$$\beta_1 = \frac{1}{n\sigma^3} \sum_{i=1}^{n} (x_i - \bar{x})^3$$

- Kurtosis de una muestra mide el cambio de pendiente de la distribución cerca del valor medio (aparición de picos):

$$\beta_2 = \frac{1}{n\sigma^4} \sum_{i=1}^{n} (x_i - \bar{x})^4$$

Estimadores de poblaciones y muestras:

PREGUNTA: Qué siginificado real tiene:

$$M_v = 18{,}5 \qquad\qquad M_v = 18{,}5 \pm 0{,}2$$

- El primer valor no tiene ningún significado.
- El segundo valor NO indica que los valores posibles de M_v se encuentran entre 18,3 y 18,7 sino que la población de donde se obtuvo la variable tiene una distribución con $\mu = 18{,}5$ y $\sigma = 0{,}2$.
- En el segundo caso se suele asumir una distribución normal debido a la aplicación del Teorema Central del Límite, pero esto no es necesariamente cierto.

Teorema central del límite:

Supongamos que tenemos una muestra $y_1,\ y_2,\ \cdots,\ y_n$ de una variable independiente que proviene de una cierta población (con media μ y varianza σ^2).
Si $Y_k = y_1 + y_2 + \cdots + y_k$, con $k \leq n$, la variable:

$$Z_k = \frac{Y_k - k\mu}{\sigma\sqrt{k}}, \quad o \quad \bar{Z}_k = \frac{(\bar{Y}_k - \mu)\sqrt{k}}{\sigma}$$

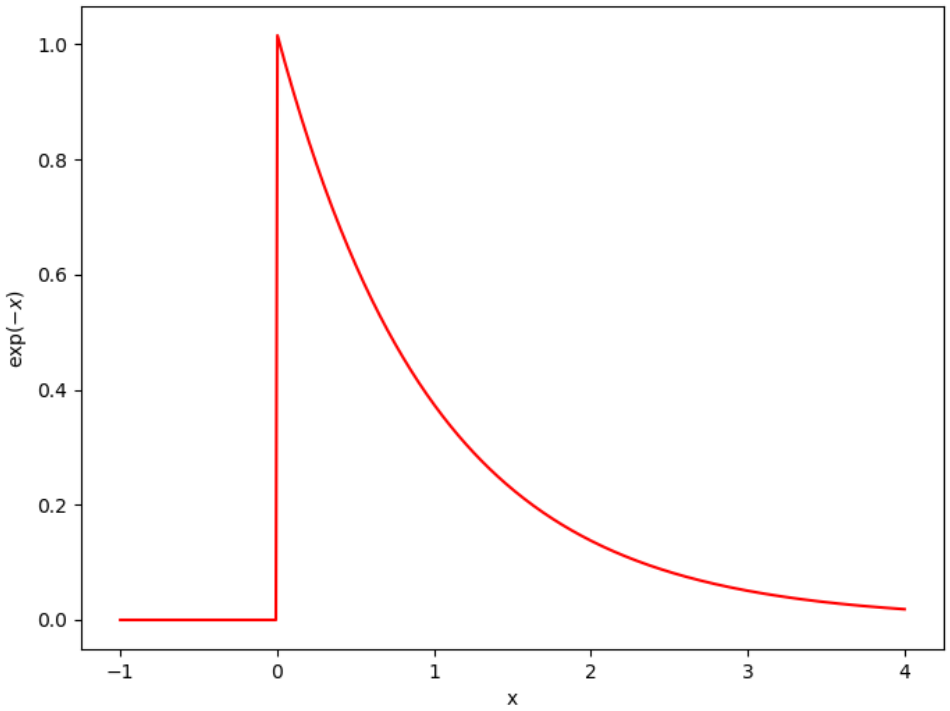
es asintóticamente normal con media 0 y variancia 1.

Notar que no es importante la forma de la distribución de la población de la cual se obtuvo la muestra !!

Teorema central del límite:

Ejemplo: supongamos que la población de donde se obtendrán los valores tiene una distribución de frecuencias que es a una exponencial truncada:

$$y = \begin{cases} 0 & , \quad x \leq 0 \\ \exp(-x) & , \quad x > 0 \end{cases}$$



Teorema central del límite:

La distribución de frecuencias para la población y su distribución acumulativa se construyen con:

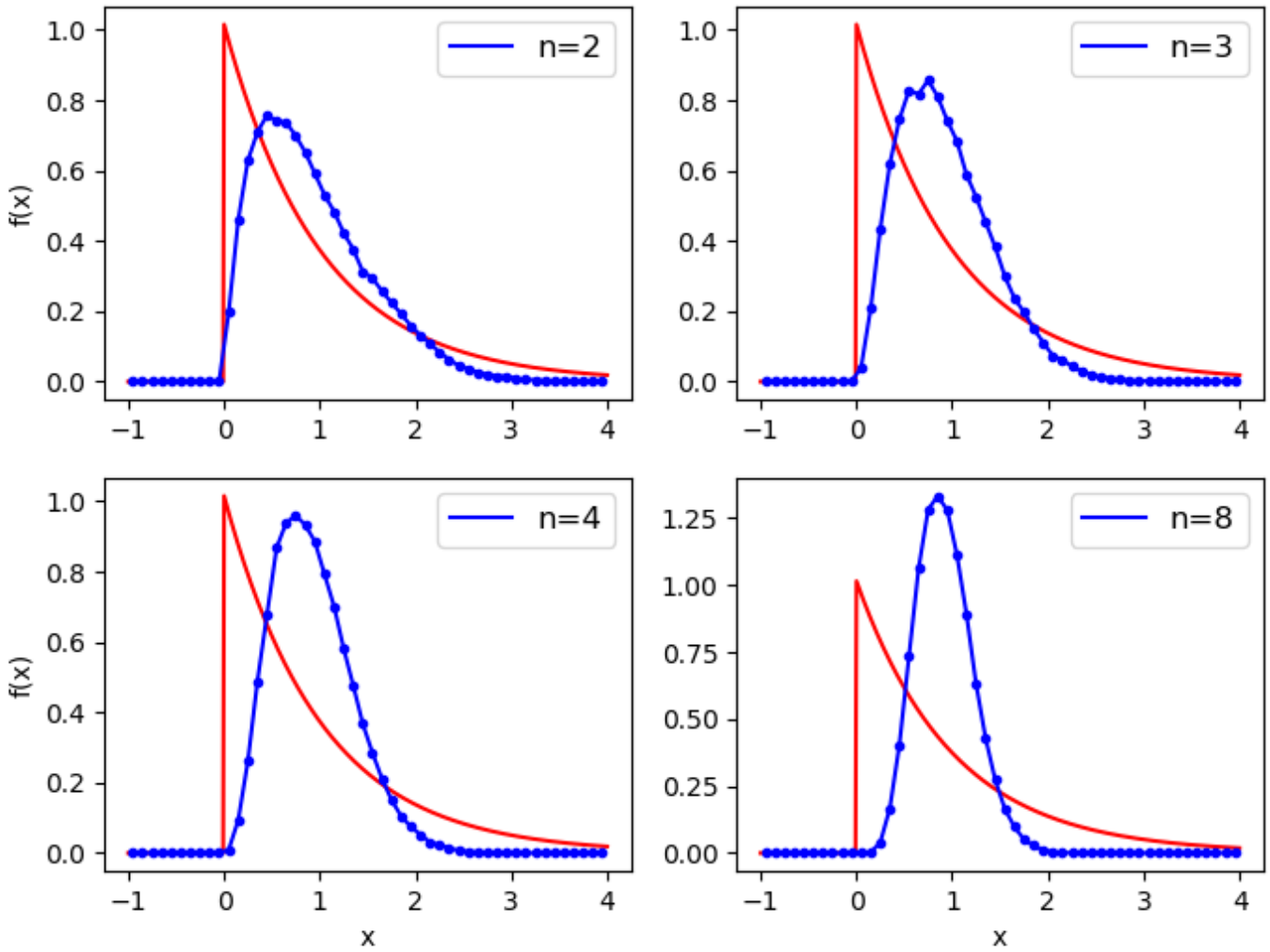
```
xx=np.linspace(-1,4,500)
yy=np.zeros(len(xx))
for ii in range(len(yy)):
 if(xx[ii]>0.):
   yy[ii]=np.exp(-xx[ii])

yy=yy/np.sum(yy)

ac=np.cumsum(yy,dtype=float)
```

Teorema central del límite:

Para obtener una muestra de p elementos promediando n valores tomados al azar de la distribución de la población y obtener su histograma hacemos una función:

```
def prom(x,ac,n,p):
  """
  Extrae p elementos promediando n valores del array x tomados al
azar de una distribucion acumulativa de frecuencias ac, y obtiene
el histograma correspondiente
  """
  vx=[]
  for ii in range(p):
      kk=np.random.random(n)
      sx=0.
      for jj in range(n):
          vp=0
          while(kk[jj] > ac[vp]):
            vp+=1
          sx+=x[vp]
      vx.append(sx/float(n))
  vx=np.array(vx)
  his=np.histogram(vx,bins=50,range=(-1,4),density=True)
  return his
```

Teorema central del límite:



Estadísticos en Python:

En Numpy se pueden encontrar las siguientes funciones tanto para arrays como para uno cualquiera de sus ejes, y con diferentes grados de libertad:

- np.mean() calcula el valor medio aritmético.
- np.average() calcula el valor medio aritmético con pesos.
- np.median() calcula la mediana.
- np.var() calcula la varianza.
- np.std() calcula la desviación standard.
- np.percentile() calcula el valor correspondiente a un percentil dado. Util para definir cuartiles.
- np.ptp() calcula el valor del rango (diferencia entre el máximo y el mínimo).

Estadísticos en Python:

Scipy también tiene funciones estadísticas tanto para arrays como para uno cualquiera de sus ejes y con diferentes grados de libertad:

- scipy.stats.describe() calcula varios estadísticos.
- scipy.stats.gmean() calcula el valor medio geométrico.
- scipy.stats.mode() calcula el modo.
- scipy.stats.kurtosis() calcula la curtosis.
- scipy.stats.skew() calcula la asimetría.
- scipy.stats.moment() calcula los momentos:

$$m_k = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^k$$

Distribuciones en Python:

En np.random hay numerosas distribuciones estadísticas disponibles, tanto para una como para varias variables.
La mayor colección de distribuciones de todo tipo está en el módulo scipy.stats.

```
Standard distributions
======================================================
standard_cauchy      Standard Cauchy-Lorentz distribution.
standard_exponential Standard exponential distribution.
standard_gamma       Standard Gamma distribution.
standard_normal      Standard normal distribution.
standard_t           Standard Student's t-distribution.
```

Distribuciones en Python:

```
Univariate distributions
============================================================
beta                 Beta distribution over ``[0, 1]``.
binomial             Binomial distribution.
chisquare            :math:`\chi^2` distribution.
exponential          Exponential distribution.
f                    F (Fisher-Snedecor) distribution.
gamma                Gamma distribution.
geometric            Geometric distribution.
gumbel               Gumbel distribution.
hypergeometric       Hypergeometric distribution.
laplace              Laplace distribution.
logistic             Logistic distribution.
lognormal            Log-normal distribution.
logseries            Logarithmic series distribution.
negative_binomial    Negative binomial distribution.
noncentral_chisquare Non-central chi-square distribution.
noncentral_f         Non-central F distribution.
normal               Normal / Gaussian distribution.
pareto               Pareto distribution.
poisson              Poisson distribution.
power                Power distribution.
rayleigh             Rayleigh distribution.
triangular           Triangular distribution.
uniform              Uniform distribution.
vonmises             Von Mises circular distribution.
wald                 Wald (inverse Gaussian) distribution.
weibull              Weibull distribution.
zipf                 Zipf's distribution over ranked data.
```

Cálculo de la moda:

La moda de una muestra se determina usualmente asignando el valor central del bin de su histograma con el mayor número de elementos, pero a veces ese procedimiento no da un buen resultado.
Supongamos que tenemos que estimar con la moda el valor del cielo para hacer fotometría y tenemos 10 pixeles con los valores:

```
In [35]: sky=np.random.random(10)*1000

In [36]: sky
Out[36]:
array([168.68303739,  63.08864929, 191.78184635, 682.20642541,
       796.88826276, 718.65806775, 726.45599338, 698.72905359,
       133.838899  , 523.36370467])
```

Cálculo de la moda:

Pero en estos casos no es una solución posible:

```
In [43]: vmax=np.max(sky)

In [44]: vmin=np.min(sky)

In [45]: vstd=np.std(sky,ddof=1)

In [46]: int((vmax-vmin)*10**(1./3.)/(3.5*vstd)+0.5)
Out[46]: 2

In [47]: np.histogram(sky,bins=2,range=(60,800))
Out[47]: (array([4, 6]), array([ 60., 430., 800.]))

In [48]: 10**0.5
Out[48]: 3.1622776601683795

In [49]: np.histogram(sky,bins=3,range=(60,800))
Out[49]:
(array([4, 1, 5]),
 array([ 60.        , 306.66666667, 553.33333333, 800.        ]))
```

Cálculo de la moda:

Un método posible está explicado en Press et al.:

1. ordenar los N valores de menor a mayor.
2. seleccionar una ventana de análisis J sobre la cual la función de distribución se va a suavizar ($J > 3$).
3. calcular un estimador con:

$$p\left(\frac{1}{2}[x_i + x_{i+J}]\right) \approx \frac{J}{N(x_{i+J} - x_i)}, \quad para \;\; i = 1, \cdots, N-J$$

4. tomar como la moda el valor $(x_i + x_{i+J})/2$ con la estimación del mayor valor.
5. la desviación standard de la moda estimada será:

$$\sigma[p(x)] \approx p(x)/\sqrt{J}$$

Cálculo de la moda:

```
import numpy as np

def estima_moda(vec,ww):
  """
  Funcion para estimar la moda de un conjunto de valores los cuales
incluso pueden ser reales. En la salida se da un tuple con el valor
estimado, la probabilidad y su desviacion standard, y el array de todos
los valores estimados.

La ventana ww debe ser preferentemente mayor a 3.
  """

  nn=len(vec)

# copio el vector de valores
#
  aa=np.copy(vec)

# ordeno el vector usando seleccion
#
  seleccion(aa)

# calculo valores de moda y sus estimaciones
#
  pp=np.zeros((nn-ww,3))
  for ii in range(nn-ww):
    pp[ii,0]=0.5*(aa[ii]+aa[ii+ww])
    pp[ii,1]=float(ww)/float(nn)/(aa[ii+ww]-aa[ii])
    pp[ii,2]=np.sqrt(float(ww))/float(nn)/(aa[ii+ww]-aa[ii])

  vmax=np.argmax(pp[:,1])
  return (pp[vmax,:]),pp
```

Cálculo de la moda:

```
In [50]: %run moda-continua.py

In [51]: moda,arr=estima_moda(sky,3)

In [52]: moda
Out[52]: array([7.04331209e+02, 6.77972721e-03, 3.91427733e-03])

In [53]: arr
Out[53]:
array([[1.27435248e+02, 2.33112555e-03, 1.34587596e-03],
       [3.28601302e+02, 7.70169180e-04, 4.44657383e-04],
       [4.25444731e+02, 5.84199293e-04, 3.37287619e-04],
       [4.45255450e+02, 5.91777597e-04, 3.41662955e-04],
       [6.21010886e+02, 1.53614265e-03, 8.86892371e-04],
       [7.04331209e+02, 6.77972721e-03, 3.91427733e-03],
       [7.47808658e+02, 3.05625934e-03, 1.76453215e-03]])

In [54]:
```

Actividades:

Práctica 5:

- Para una variable cualquiera de su archivo de datos calcule todos sus estadísticos.
- Para esa misma variable verificar si el porcentaje de valores dentro de $1\,s$, $2\,s$ y $3\,s$ de $\bar{x}$ representan el $\sim 68{,}3\,\%$, $\sim 95{,}5\,\%$ y $\sim 99{,}7\,\%$ de la muestra, respectivamente.
- Para esa misma variable extraiga al azar 20 muestras con 100 elementos cada una, calcule los estadísticos de cada una de ellas y compare con los valores obtenidos en el primer punto.
- Los valores medios de las muestras usadas en el ejemplo para demostrar el Teorema del Límite central (pág. 24) tienden a un valor de 0,9228. Por qué?.

Actividades:

Práctica 5 (cont.):

- Utilizando la función de luminosidad de Schechter, $\phi(L) = \left(\frac{\phi^*}{L^*}\right)\left(\frac{L}{L^*}\right)^{\alpha} \exp(-L/L^*)$, con $L^* = 1$, $\phi^* = 1$ y $\alpha = -0{,}5$, generen una población de galaxias para $0 < L < 6$ y extraigan muestras de 10 galaxias. Cómo se distribuye el valor máximo de cada muestra?. Y en el caso de extraer 50 galaxias?.

Entrega

Para la próxima clase

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