Programación Numérica para Geofísica

PNG

Andrés Sepúlveda

Departamento de Geofísica
Universidad de Concepción

15 Junio 2020

Estadística

Funciones más comunes

mean	median	mode
skewnes	kurtosis	meansq
std	var	statistics

mean - Promedio

```
aux = mean(x,dim,opt)
```

$$mean(x) = \frac{\sum_i x_i}{n}$$

Si x es una matriz, entrega el resultado por columna.
opt: 'a' - media aritmética
opt: 'g' - media geométrica

median - mediana

```
                 x(ceil(N/2))                - N impar
median(x,dim) =
                 (x(N/2) + x((N/2)+1))/2  - N par
```

Si los elementos están ordenados, indica cual está a la mitad.

mode - moda

`mode(x,dim)`

Presenta el valor mas frecuente.
En una distribución Gaussiana, media = moda = mediana.

Skewness

```
skweness(x,flag,dim)
```

$$skewness(x) = \frac{mean((x - mean(x))^3)}{std(x)^3}$$

Inclinación de la curva
Derecha - Negativo
Izquierda - Positivo

Kurtosis

`kurtosis(x,flag,dim)`

$$kurtosis(x) = \frac{mean((x - mean(x))^4)}{std(x)^4}$$

flag - corrige sezgo
Indica lo aplastado o elongado de la distribución.

¿Alguna tendencia en la fórmula?

meansq - Promedio de los cuadrados

`meansq(x,dim)`

$$meansq(x) = \frac{1}{N}\sum_{i} x(i)^2$$

std - Desviación estandard

`std(x,opt,dim)`

$$std(x) = \sqrt{\frac{1}{(N-1)} \sum_{i} (x(i) - mean(x))^2}$$

opt: 1/0 - usar N-1 o N

Piensen en cantidad de información
Grados de Libertad

var - Varianza

`var(x,opt,dim)`

$$var(x) = \frac{1}{(N-1)} \sum_i (x(i) - mean(x))^2$$

opt: $1/0$ - usar N-1 o N

statistics - tutti

```
statistics(x,dim)
```

Todos los anteriores, y más...

cov - Matriz de Covarianza

`cov(x,y,opt)`

$$cov(x,y) = \frac{1}{(N-1)} \sum_i ((x(i) - mean(x)) \times (y(i) - mean(y)))$$

opt: $1/0$ - usar N-1 o N

Estadística

Funciones auxiliares

isnan	isinf	isfinite
find	range	iqr
nonzeros		

isnan -Identifica posición de los NaN

```
x = [NaN,1,2,nan,4,5];
y = isnan(x);

a = [ 1 2 ; 3 NaN];
b = isnan(a);

a(b==1)=4;
```

isinf -Identifica posición de los “infinitos“ (inf o Inf)

```
c = 1/0

x = [Inf,1,2,inf,4,5];
y = isinf(x);

a = [ 1 2 ; 3 inf];
b = isinf(a);

a(b==1)=4;
```

isfinite -Identifica posición de los “finitos“ (no Nan o Inf)

```
x = [Inf,1,2,inf,4,5];
y = isfinite(x);

a = [ 1 2 ; 3 NaN];
b = isfinite(a);

a(b==1)=NaN;
```

find(x,N,direction) - ubica los (N) (primeros/últimos) elementos que no son cero en un vector/matriz

```
find(eye(2))                      % Entrega un vector

[i j]=find(3 * eye(3))            % Entrega los indices
```

Podemos "inducir" ceros en un vector/matriz al definir una condición con $>, <, ==$

```
lon=-180:1:180;

area5=find(lon > -120 & lon < -69);
```

find es su amigo

range - Diferencia entre el máximo y el mínimo

```
x=rand(100,1);

y=range(x)
```

No es **robusto**

iqr - Rango entre los dos primeros cuartiles (50%)

```
x=rand(100,1);

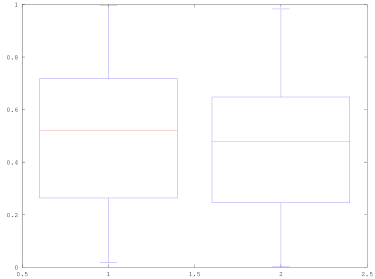
y=iqr(x)
```

robusto

Es la base del famoso **boxplot**

Boxplot - Centro, 25%, 75%, valores extremos (1.5-3.0 IQR, $>$3 IQR)

```
x=rand(100,2);
boxplot(x);
```



nonzeros - Entrega un vector con los elementos que no son cero

```
x = eye(3);

y = nonzeros(x)
```

La utilidad de varias de estas funciones radica en crear 1's o 0's usando condiciones $(>, <, ==')$

Operadores Relacionales

Octave

- a == b
- a < b
- a > b
- a <= b
- a >= b
- a ~= b

Operadores Lógicos

Octave

- a && b
- a || b
- a &b / and(a,b)
- a | b / or(a,b)
- xor(a,b)
- not(a)

Raíces y Logaritmo

Matlab

- sqrt(a)
- log(a)
- log10(a)
- log2(a)
- exp(a)

Redondeo

Matlab

- round(a)
- ceil(a)
- floor(a)
- fix(a)

- Definir los parámetros de la mejor recta que describe la relación entre dos series de datos (X e Y) es un problema muy importante en Geofísica, y muy recurrente.
- "*No busques una solución no-lineal, busca la mejor aproximación lineal a tus problemas*"

  (Un físico famoso a otro)
- La clave del asunto está en definir cuantitativamente "mejor".
- Un problema interesante si consideramos que, además de X e Y, tenemos el error de X y el error de Y.

- La primera solución que todos deben manejar es: **polyfit**

```
polyfit (X, Y, 1)
P=polyfit (X, Y, 1);
```

- ```
  octave:1> x=1:10;
  octave:2> y = 5*x+8 ;
  octave:3> polyfit(x,y,1)
  ans =   5.0000   8.0000
  ```

- Con "ruido" (¿Qué es ruido?)

  ```
  octave:4> y = 5*x+8 + rand(10,1)';
  octave:5> polyfit(x,y,1)
  ans =   5.0318   8.2589
  ```
  ```

- Guardando los parámetros como variables

```
octave:6> P = polyfit(x,y,1);
octave:6> yfit = P(1)*x + P(2);
octave:6> yfit = polyval(P,x);
```

- ¿Suficiente?
- Falta considerar:
  1. Índices estadísticos de calidad del ajuste.
  2. El error asociado a los parámetros calculados.
  3. La influencia del error en Y.
  4. El método de ajuste.
  5. La influencia del error en X.

- En el punto 1) el indicador más usado es el coeficiente de correlación

$$r_{xy} = \frac{\overline{x \cdot y} - \overline{x} \cdot \overline{y}}{\sqrt{(\overline{x^2} - \overline{x}^2)(\overline{y^2} - \overline{y}^2)}}$$

- A mayor error de un dato, menor "peso" (*weight* -> w) tiene en el ajuste.
- Existen métodos no robustos (OLS: *ordinary least squares*), y metodos robustos (IRLS: método iterativo)
- A través de la matriz de covarianza (o métodos Monte Carlo) se puede obtener un intervalo de confianza para m y n. *Ver el código*.

El tema puede ser 50 veces mas complejo, pero en orden de prioridad

- Obtengan m y n. Sepan usar **polyfit**. (y **polyval**)
- Obtengan el coeficiente de correlación
- Obtengan el error asociado a m y n.
- Sepan como considerar el error de Y.
- Sepan como lidiar con los valores extremos (regresión robusta).

- Observen que la ecuación

$$y = m \cdot x + n$$

es distinta a

$$y = m \cdot x$$

- El concepto de regresión lineal tambien cubre los siguientes casos

$$y = m \cdot x^2 + n \cdot x + p$$

y

$$z = A \cos \omega \cdot t + B \sin \omega \cdot t$$

(¿Por qué?)

- Mucha mas información en:

`http://reliawiki.org/index.php/Simple_Linear_Regression_Analysis`