

Po častiach lineárna empirická distribučná funkcia

PČLEDF zobrazuje podrobne to, čo histogram, a preto jej konštrukcia má význam skôr pre menší počet spracovávaných údajov. Navyše predpokladáme, že údaje nie sú kompletne a predstavujú určitú reprezentatívnu vzorku celej skúmanej reality.

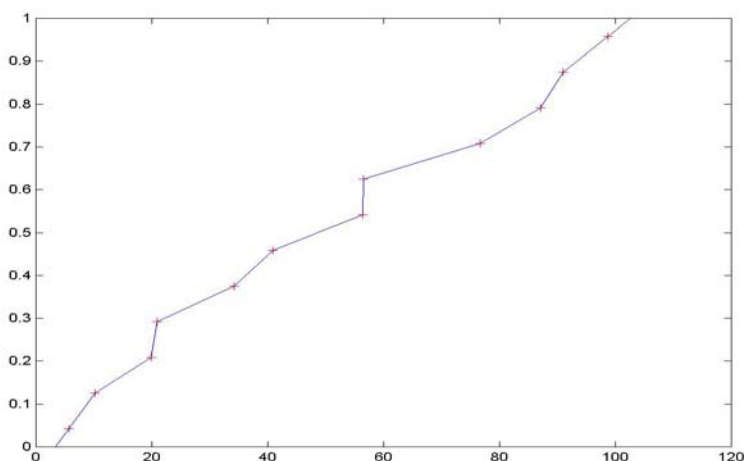
```
» x=sort(rand(1,12)*100);
```

Vo vektore x máme už usporiadaných 12 hodnôt. Tie by sa mali viazať s nasledujúcimi hodnotami intervalu (0,1):

```
» h=1/12; y=h/2:h:1;  
» plot(x,y)
```

Obrázok je ešte neúplný, potrebujeme ho doplniť, aby čiara začínala od nuly a prišla až po 1.

```
» n=length(x); x0=(3*x(1)-x(2))/2; xz=(3*x(n)-x(n-1))/2;  
» xx=[x0,x,xz]; yy=[0 y 1];  
» plot(x,y,'+r'), hold on, plot(xx,yy)
```



Hľadanie kvantilov (definovaných cez PČLEDF) umožňuje funkcia *interp1*, ktorá pospája zadané body rovnými úsečkami a odpovie na otázku o funkčnej hodnote ktoréhokoľvek bodu intervalu. Konkrétne v našom prípade príkaz `interp1(yy,xx,q)` spôsobí, že Matlab bude mať na mysli vyššie uvedený graf (zadali sme yy, xx v opačnom poradí, lebo q bude číslo z y-ovej osi), nájde si hodnotu q na y-ovej osi a pozrie sa, pre aké x naša pčledf dosahuje hodnotu q.

Nájdime najprv medián, kvartily a body, v ktorých graf „dosadne“ na 0 a 1:

```
» interp1(yy,xx, [0, 0.25, 0.5, 0.75, 1])
```

```
ans = 3.3799 20.4165 48.6367 81.9202 102.5306
```

Úloha 1: Nájdite a-kvantil, kde $a = 0.05, 0.15, 0.3, 0.6, 0.85, 0.99$.

Úloha 2: Kreslite PČLEDF a počítajte rôzne kvantily pre iné súbory hodnôt.

Spracovanie dátového súboru

Naším cieľom je zistiť, aké hodnoty sú prítomné vo vektore dát v , usporiadať ich podľa veľkosti, zistiť ich početnosti a kumulatívne početnosti. Je viacero spôsobov, ako sa to dá urobiť, preto môžete uvažovať aj o iných postupoch s využitím iných príkazov.

Pre názornosť budeme postup ilustrovať na príklade. Kvôli názornosti nechávame všetky výstupy vypisovať. Ak budete tento postup (po jeho pochopení) používať pri väčšom súbore dát, nezabudnite doplniť bodkočiarky tam, kde treba.

Vstupné dáta:

```
>> v=round(rand(1,23)*40-20)
v =
 18 -11  4 -1 16 10 -2 -19 13 -2  5 12 17 10 -13 -4 17 17 -4 16 -18 -6 13
```

Usporiadanie:

```
>> x=sort(v), n=length(x)
x =
-19 -18 -13 -11 -6 -4 -4 -2 -2 -1  4  5 10 10 12 13 13 16 16 17 17 17 18
n = 23
```

Číselné zmeny – skúste z formulácie príkazu a odpovede naň zistiť, čo to robí:

```
>> d=[1, x(1:n-1)<x(2:n)]
d =
 1  1  1  1  1  1  0  1  0  1  1  1  1  0  1  1  0  1  0  1  0  0  1
```

Pozície, kde sa v súbore x mení číselná hodnota:

```
>> w=find(d)
w =  1  2  3  4  5  6  8 10 11 12 13 15 16 18 20 23
```

Zoznam a počet rôznych hodnôt prítomných vo vektore x :

```
>> y=x(w), m=length(y)
y = -19 -18 -13 -11 -6 -4 -2 -1  4  5 10 12 13 16 17 18
m = 16
```

Početnosť jednotlivých hodnôt v zozname y – dobre sa zamyslite nad dole uvedeným príkazom a skúste pochopiť, čo robí:

```
>> p=[w(2:m), n+1] - w
p =  1  1  1  1  1  2  2  1  1  1  2  1  2  2  3  1
```

Poznámka:

Rovnaký výsledok dostaneme aj pomocou príkazu `histc`, bez akýchkoľvek ďalších úprav. Tu sa ukazuje, aký význam má skutočnosť, že `histc` uvádza zvlášť počet prvkov rovných poslednej hranici.

```
>> ph=histc(x,y)
```

```
ph = 1 1 1 1 1 2 2 1 1 1 2 1 2 2 3 1
```

Kumulatívne početnosti:

```
>> c=cumsum(p)
```

```
c = 1 2 3 4 5 7 9 10 11 12 14 15 17 19 22 23
```

Prehľadný výpis nakoniec:

```
>> [y' p' c']
```

```
ans =
```

```
-19 1 1  
-18 1 2  
-13 1 3  
-11 1 4  
-6 1 5  
-4 2 7  
-2 2 9  
-1 1 10  
4 1 11  
5 1 12  
10 2 14  
12 1 15  
13 2 17  
16 2 19  
17 3 22  
18 1 23
```

Úloha: Urobte podobnú analýzu pre súbor

```
>> v=round(rand(1,193)*100-30)
```