

## Kvantily II

### Počítanie kvantilov pri veľkých triedených súboroch dát.

Je daný vektor dát

```
>> x=rand(1,100000).^5*100-50;
```

a hranice tried

```
>> t=0:10; ht=t.^2-50
```

```
ht = -50 -49 -46 -41 -34 -25 -14 -1 14 31 50 .
```

Zistíme početnosti a (relatívne) kumulatívne početnosti jednotlivých tried

```
>> pt=histc(x,ht); kpt=cumsum(pt); rkpt=kpt/100000
```

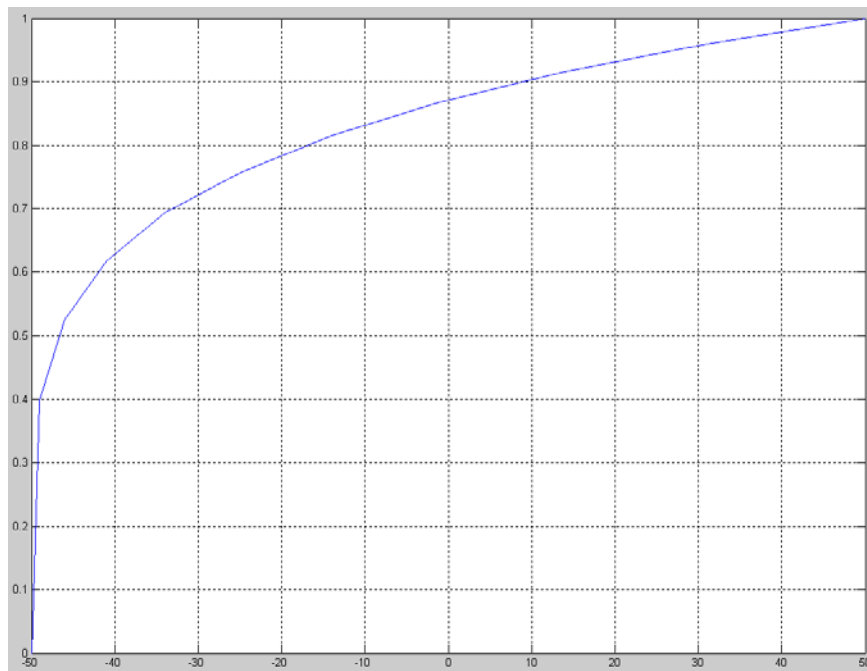
a zabudneme (!!!) na vektor *x* (`clear x`). Sme v situácii, keď sú dané triedy a ich početnosti, ale o samotných prvkoch už nevieme nič (keby sme vedeli, môžeme voliť presnejší postup ako ten, ktorý nasleduje).

Posledný údaj v *rkpt* je totožný s predposledným, môžeme ho teda ignorovať (ak by nebol, musíme ho pripočítať k predposlednému). Na začiatok pridáme nulu.

```
>> rkpt=[0, rkpt(1:10)]
```

Nakreslíme si zodpovedajúci polygón:

```
>> plot(ht,rkpt), grid on
```



Tento polygón nám pomôže počítať kvantily. Využijeme ML-funkciu *interp1*, ktorá si (sama pre seba akoby) pospája zadané body rovnými úsečkami, teda predstaví si taký polygón, aký sme nakreslili, a odpovie na otázku o funkčnej hodnote ktoréhokoľvek bodu intervalu. Napríklad príkaz *interp1(rkpt,ht,q)* spôsobí, že Matlab nájde hodnotu *q* na y-ovej osi (preto zadávame *rkpt* ako prvé) a pozrie sa, pre aké *x* náš graf dosahuje hodnotu *q*.

Nájdime najprv medián a kvantily:

```
» interp1(rkpt, ht, [0.25, 0.5, 0.75])
```

```
ans = -49.37219055272344 -46.60249666326450 -26.04740157480315
```

Takýmto spôsobom sa môžeme pýtať aj na iné kvantily, napr. 0,95-kvantil:

```
>> interp1(rkpt, ht, 0.95)
```

```
ans = 2.710594965675055e+001
```

Môžeme sa pýtať aj naopak – hodnota 0 zodpovedá akému kvantilu?

```
>> interp1(ht, rkpt, 0)
```

```
ans = 8.713140000000000e-001
```

Číslo 0 predstavuje zhruba 0.87-kvantil v danom súbore triedených dát.

### Úloha:

Zvoľte si náhodný (ale podľa veľkosti usporiadaný) vektor *ht* dĺžky 1001 a podobne náhodne nájdite vektor početností tried daných hranicami *ht*. Počítajte rôzne kvantily.