

## Računalniška orodja v fiziki

18. marca 2009

### 2. tema - Histogrami

Histogrami so grafični prikazi, pri katerih je ključna ena sama spremenljivka, druga ni podana ali pa je definirana na eni od standardnih množic (zaporedna številka, dan v mesecu, letnica). Posebno pogosti so v ekonomskih in družbenih dejavnostih (n.pr. cena sodčka nafte Brent v zaporednih mesecih leta; število prometnih prekrškov v določenem obdobju, prikazano po regijah, itd.), kjer imajo stalno podobo stolpičastega grafa. V fiziki nam mnogokrat zadošča stopničasti graf, torej črta konstantne višine nad zaporednimi intervali abscisne spremenljivke. Seveda lahko uporabimo tudi standardne označevalce točk v grafih (pike, krožce, zvezdice), vendar ima stopničasta črta majhno vsebinsko prednost, kot bomo videli pozneje.

V fiziki je histogram največkrat prikaz pogostnosti spremenljivke. Merimo količino, ki se od meritve do meritve naključno spreminja, ne da bi te spremembe sami s čim povzročili. Datoteka »Agxx.dat« podaja število radioaktivnih razpadov šibkega izotopa, ki smo jih z detektorjem zaznali v zaporednih sekundah. Četudi je neodvisna spremenljivka – čas – znana, je nepomembna, ker se vrednosti ne spreminjajo sistematično s časom. Zato je datoteka enostolpčna, časa ni treba pisati, ker ga ob branju lahko rekonstruiramo iz vrednosti števil vsake točke. Lahko torej narišemo časovni graf, vendar z njim ne bomo nič pametnejši. V resnici je cilj meritve kvečjemu kaka povprečna vrednost, o čemer bomo govorili v enem od naslednjih projektov. Danes pa si lahko ogledamo graf porazdelitve po številu razpadov. Podatke  $y[j]$ ,  $j = 1, \dots, n$ , ki jih preberemo iz datoteke, je treba najprej »opredalčiti« (»binning«). Po postopku iz prejšnjega projekta si najprej paščemo meji spremenljivke:  $y_{\min}$  in  $y_{\max}$ . Z njima uvedemo register `numy`:

```
array: numy[ymin..ymax] of integer;
```

```
for i := ymin to ymax do numy[i] := 0;
```

V predalčke registra spravljamo podatke enegaza drugim:

```
for j := 1 to n do inc(numy[y[j]]);
```

potem pa graf `numy[i]` narišemo. Vidimo, da obstaja število, ki je največkrat zastopano, tudi temu bližnji predalčki so še dobro zastopani, bolj ko se oddaljujemo k večjim ali manjšim vrednostim, pa število zadetkov vse bolj pada, dokler ne pade na 0. – Če števila `numy` še delimo s številom vseh podatkov  $n$ , dobimo normirano porazdelitev, ki v vsakem predalčku pove, v kolikem deležu se pojavlja določena vrednost.

V gornjem primeru z izbiro velikosti predalčkov ni bilo težave, spremenljivka je sama narekovala njihovo velikost. Drugače je, če imamo spremenljivko tipa `real`, kjer velikost predalčka ni samoumevna pač pa jo izberemo glede na cilj prikaza. Tudi ne bodo v predalčku več same enake vrednosti, pač pa vrednosti v intervalu med  $y[i]$  in  $y[i]+h$ , kjer je  $h$  širina predalčka.

Pripravimo si datoteko števil  $y[i]$

```
for i:= 1 to 628 do y[i]:= sin(0.1*i);
```

Če narišemo  $y$  v odvisnosti od  $i$ , bomo seveda dobili deset valov sinusoide. Zdaj pa vrednosti uredimo v predalčke. Koliko naj jih bo? Zavedati se moramo, da je pri naključnem zbiranju razsutost v predalčku enaka  $m \pm \sqrt{m}$ . Naše vrednosti se nahajajo na intervalu  $[-1, 1]$ . Pri  $h=0.1$ , torej pri 20 predalčkih, bo v enem predalčku v povprečju 31 vrednosti, v resnici pa bo to število razsuto za tipično  $\pm\sqrt{31} = \pm 5$ , se pravi, za šestino. To pomeni, da bo naš histogram kar precej zobat. Če bi hoteli bolj gladek histogram, bi morali vzeti bolj grobo delitev (večje predalčke) ali še boljše, več števil.

Izberemo  $h$ .

```
var ii: integer;
```

```
m := (ymax-ymin)/h;
```

```
for i := 1 to m do numy[i] := 0;
```

```
for j := 1 to n do begin
```

```
  ii := min(int((y[i] - ymin)/h)+1,m); inc(numy[ii])
```

```
end;
```

V sliki je seveda treba narisati vsak  $numy$  pri vrednosti  $ymin + h*ii$ , ne pa pri samem  $ii$ .

V orodju **Excel** lahko postopek za predalčenje zložimo iz menujskih izbir: tabelo vrednosti uredimo po velikosti (sort) in uvedemo rang. To je zaporedna številka, ki pa ostaja enaka za enake vrednosti. Razlika dveh zaporednih rangov je iskani  $numy$ . V **Mathematici** je risanje histogramov podprto. Najprej naračunamo ali pa iz datoteke preberemo enodimenzionalno tabelo s podatki. Histogram narišemo s *Histogram[ $podatki$ ]*, kjer je *podatki* tabela podatkov. Dodatne nastavitve so opisane v pomoči. **Gnuplot** neposredno ne omogoča risanja histogramov, tako da si moramo najprej pripraviti datoteko, v kateri naj bo v prvem stolpcu koordinata predalčka, v drugem pa vrednost v tem predalčku. Histogram narišemo npr. tako: *plot "podatki" with boxes fs solid 0.5*.

Naloge:

1. Nariši histogram s podatki v datoteki »Agxx.dat«.
2. Nariši histogram s podatki iz datoteke »Ozadje.dat«. To je meritev absorpcije rentgenskih žarkov (logaritemrazmerja vpadnega in prepuščenega toka, drugi stolpec) brez merjenca, tako da pričakujemo konstantne ali skoraj konstantne vrednosti. Izberi primerno gostoto predalčenja. Ali je rezultat kaj podoben histogramu iz naloge 1?
3. Nariši histogram porazdelitve podatkov iz datoteke »Fe\_Codat«, ki vsebuje nekoliko predelan absorpcijski spekter EXAFS (Extended X-ray Absorption Fine Structure = drobna struktura rentgenskih absorpcijskih robov) mešanega železovo-kobaltovega oksida. Čeprav je neodvisna spremenljivka (energija fotona v eV) netrivialna in pomembna fizikalna količina, je zanimivo pogledati podatke tudi neodvisno od energije. Izberi primerno število predalčkov. Porazdelitev ima vrhove pri skoraj konstantnih vrednostih absorpcije med robovi. – Zanimiva je še ena modifikacija: koraki v energiji, pri katerih smo merili absorpcijo, niso enako razmaknjeni (ekvidistantni). Zato v

predalčenju vse točke niso enako pravično obravnavane. Predstavljajmo si, da opravimo meritev še enkrat z zelo drobnim ekvidistantnim energijskim korakom. Vidimo, da je pravična teža vsake točke velikost koraka, točneje interval, ki obsega pol desnega in pol levega energijskega koraka. S tem smo določili matematično **mero** vsake točke. Zadnji ukaz v gornji skici predalčenja (pred end;) zamenjamo z:

```
numj[ii] := numj[ii] + 0.5*(E[j+1] - E[j-1]);
```

Primerjaj porazdelitev po prejšnjem in novem načelu. Če je prva normirana s številom vseh točk, je druga normirana z dolžino energijskega intervala celega spektra,  $E_{\max} - E_{\min}$ .

4. Nariši histogram za sinusno funkcijo, omenjen v tekstu. Sliko lahko razumemo kot porazdelitev časa, ki ga sinusno nihajoča točka preživi v oddaljenosti  $y$  od mirovne lege.