

Stĺpcový a koláčový diagram

Stĺpcový diagram slúži na grafické znázornenie početnosti niekoľkých položiek, ktoré považujeme za rovnocenných „súperov“. V koláčovom diagrame zase názorne vyjadrujeme ich relatívne početnosti, teda podiel na celku. Typickým príkladom môže byť spracovanie volebných výsledkov.

Vo voľbách v istej krajine kandidovalo 12 strán s nasledujúcimi výsledkami:

Strany	Počet hlasov
1. Pravá strana	11235
2. Ľavá strana	17854
3. Strana zelených	9874
4. Strana fialových	5789
5. Strana slobody	14565
6. Strana disciplíny	621
7. Strana práce	1123
8. Strana oddychu	27894
9. Ľudová strana	16784
10. Snobská strana	999
11. Strana roľníkov	12345
12. Strana bezzemkov	13698

Vložíme do Matlabu potrebné údaje. Poradové čísla s rovnakým rozstupom zadávame jednoduchým príkazom `a:k:b`, kde `a`, `b` sú krajné body intervalu a `k` je krok. Výsledkom príkazu je vektor `[a, a+k, a+2*k, ... v]`, kde `v` je najväčšie číslo tvaru `a+n*k`, ktoré je ešte menšie ako `b`. Pokiaľ nezadáme krok (tj. napíšeme `a:b`), Matlab pracuje s krokom 1.

Naše poradové čísla 1 až 12 môžeme zadať takto:

```
>> porcislo=1:12;
```

Bodkočiarkou na konci zakazujeme Matlabu výpis vektora. Skúsme to bez bodkočiarky:

```
>> porcislo=1:12
```

```
porcislo =  
1 2 3 4 5 6 7 8 9 10 11 12
```

Možnosť zákazu výpisov oceníme najmä v prípade, ak ide o veľmi dlhé vektory.

Volebné výsledky musíme naložiť ručne:

```
>> vysledky=[11235 17854 9874 5789 14565 621 1123 27894 16784 999 12345 13698];
```

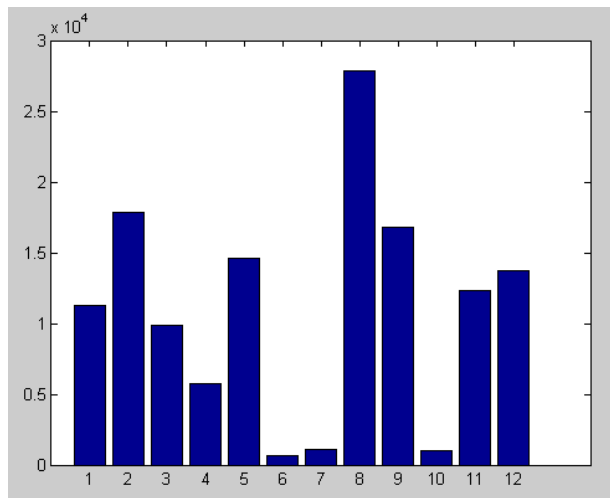
Vektory zapisujeme do hranatých zátvoriek a jednotlivé zložky oddeľujeme medzerami alebo čiarkami. V Matlabe platí duálny obeh, môžeme používať čiarky aj medzery (dokonca v rámci jedného vektora), ale Matlab „vydáva iba v medzerách“.

Stĺpcový diagram získame v Matlabe príkazom `bar`.

Vzhľadom na to, že máme strany očíslované pekne porade, stačí prikázať:

```
>> bar(vysledky)
```

a Matlab si sám doplní čísla 1 až 12:

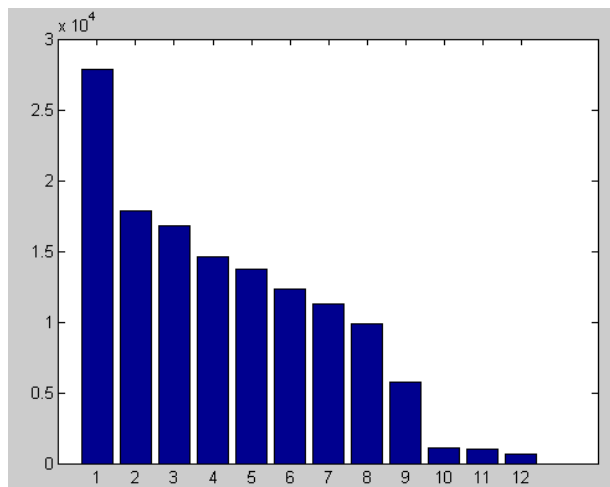


Skúsme však nakresliť histogram tak, aby strany boli usporiadané v poradí podľa úspešnosti. Najprv si ich usporiadame – na to slúži príkaz sort:

```
>> uspechy=sort(vysledky), bar(uspechy)
```

```
uspechy=
```

```
621    999    1123    5789    9874    11235    12345    13698    14565    16784    17854    27894
```



Volebné zisky sú usporiadané, ale stratili sme prehľad o tom, komu patrili. Pri 12-tich stranách si to ešte dokážeme zrekonštruovať, ale pri väčšom počte položiek by sme mali problém. Preto nás poteší, ak Matlab spolu s usporiadaním výsledkov uvedie aj poradie čísel jednotlivých strán – k tomu ho donútíme tak, že mu pripravíme dva priečinky na zapisovanie výsledkov príkazu sort. Do prvého zapíše usporiadané zložky vektora *vysledky*, do druhého ich poradové čísla v pôvodnom vektore *vysledky*:

```
>> [uspechy poradie]=sort(vysledky)
```

```
uspechy =  
    621    999    1123    5789    9874    11235    12345    13698    14565    16784    17854    27894
```

```
poradie =      6    10    7    4    3    1    11    12    5    9    2    8
```

Vo voľbách sa úspech meria počtom hlasov, preto by sa nám zišlo opačné usporiadanie oboch vektorov. Ľavoprávé otočenie vektora (aj matice) zabezpečí príkaz `fliplr` (= flip left right):

```
>> uspechy=fliplr(uspechy); poradie=fliplr(poradie);
```

Zároveň nás bude zaujímať, kto získal koľko percent z celkového počtu hlasov. Sčítame hlasy (príkaz `sum`) a sumou predelíme vektor *uspechy*:

```
>> relusp=uspechy/sum(uspechy);
```

Vypíšeme si nakoniec v stĺpcoch vedľa seba: poradie strany vo voľbách, pôvodné poradové číslo, absolútny zisk hlasov a relatívny podiel na hlasoch. Riadkový vektor zmeníme na stĺpcový (aj naopak) transpozíciou, ktorú v Matlabe vyjadríme bežným apostrofom.

```
>> [(1:12)',poradie',uspechy', relusp']
```

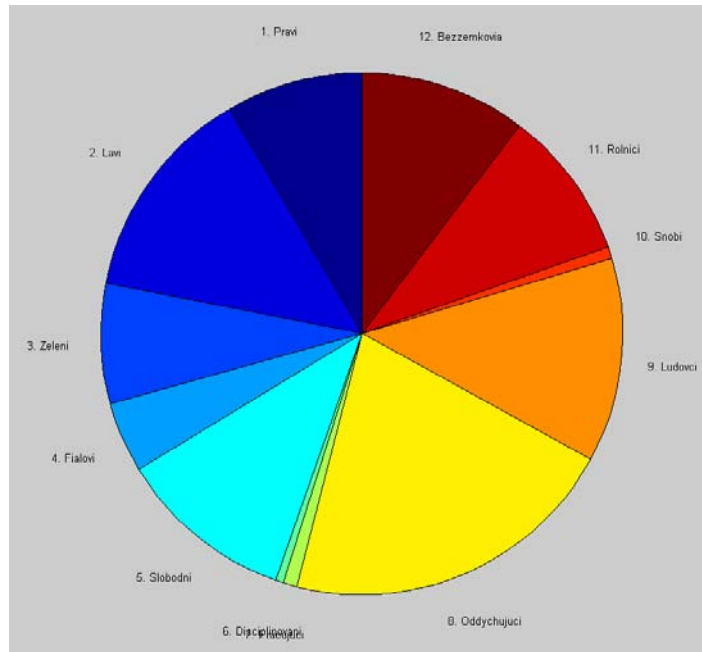
Odozva na uvedený príkaz nás asi nenadchne. Matlab nevie, čo znamenajú položky v jednotlivých stĺpcoch a mení ich na spoločný formát, ktorý je neprehľadný. Budeme preto uvádzať absolútny volebný zisk v tisícoch:

```
>> [(1:12)',poradie',uspechy'/10^3, relusp']  
ans =  
    1.0000    8.0000   27.8940    0.2101  
    2.0000    2.0000   17.8540    0.1345  
    3.0000    9.0000   16.7840    0.1264  
    4.0000    5.0000   14.5650    0.1097  
    5.0000   12.0000   13.6980    0.1032  
    6.0000   11.0000   12.3450    0.0930  
    7.0000    1.0000   11.2350    0.0846  
    8.0000    3.0000    9.8740    0.0744  
    9.0000    4.0000    5.7890    0.0436  
   10.0000    7.0000    1.1230    0.0085  
   11.0000   10.0000    0.9990    0.0075  
   12.0000    6.0000    0.6210    0.0047
```

Na zobrazenie relatívnych úspechov použijeme koláčový diagram (príkaz `pie`). Na rozdiel od stĺpcového diagramu tu máme možnosť nazvať jednotlivé podiely menom strany:

```
>> Nazvy={'1. Pravi' '2. Lavi' '3. Zeleni' '4. Fialovi' '5. Slobodni' '6. Disciplinovani'  
        '7. Pracujuci' '8. Oddychujuci' '9. Ludovci' '10. Snobi' '11. Rolnici' '12. Bezzemkovia'};  
>> pie(vysledky,Nazvy)
```

Pozn.: Obrázok bude rovnaký s argumentom `vysledky` aj `relusp`.



Aj grafika Matlabu má svoje medze...

Pomôžeme si preto známou 5% hranicou ako podmienkou na vstup do parlamentu. Vidíme, že ju spĺňa 8 strán, ale ak by sme pracovali s väčším vektorom, uvítali by sme, ak by danú hranicu uvidel aj Matlab. Naučíme ho to:

```
>> qq=find(relusp>0.05)
qq = 1 2 3 4 5 6 7 8
```

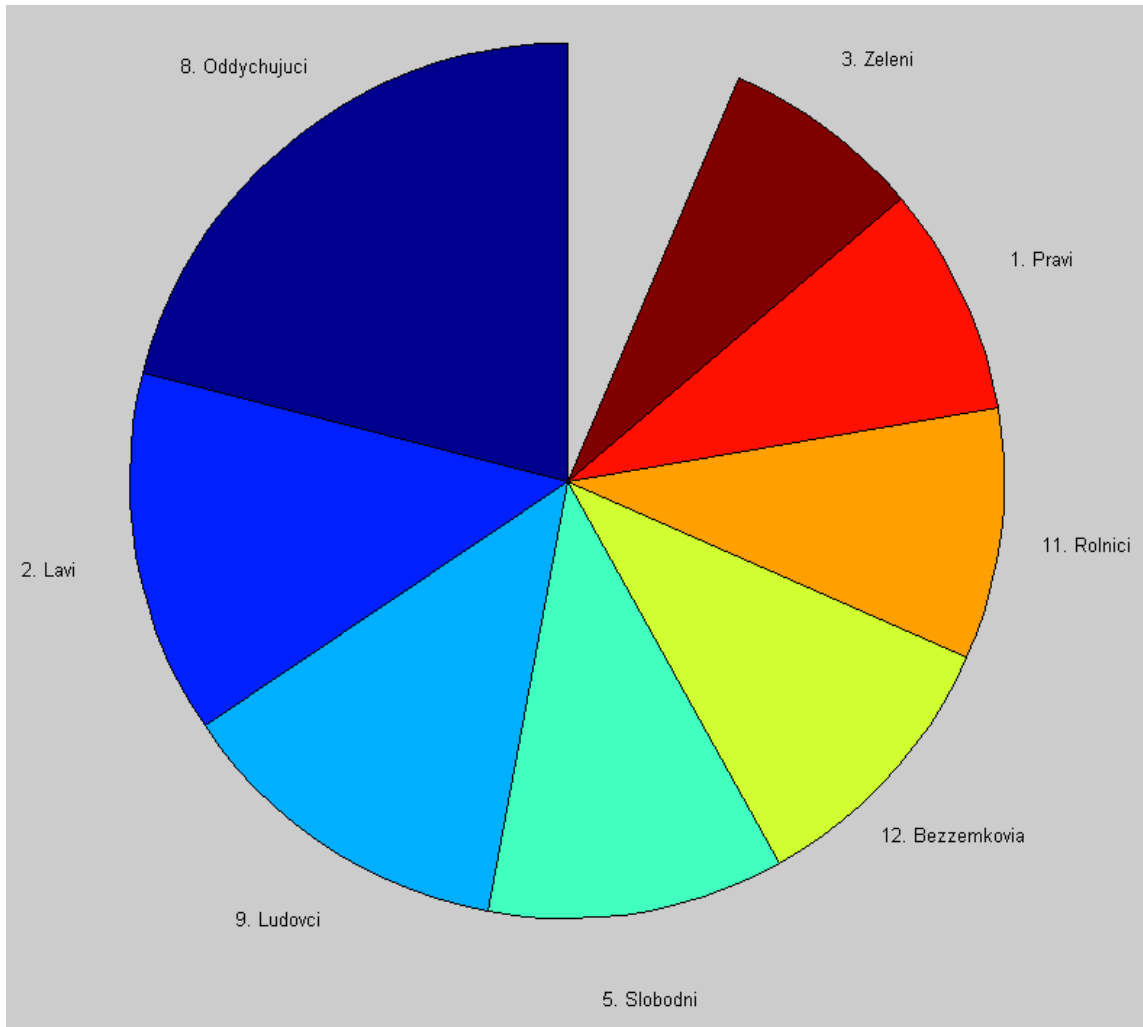
Vektor `relusp>0.05` má jednotky tam, kde je podmienka splnená, inde nuly. Príkaz `find` vypíše pozície s nenulovou hodnotou vo vektore. Pri dlhých vektoroch nie je dobré `qq` vypisovať. Vieme, že vektor `relusp` je usporiadaný podľa veľkosti, a preto to zostručíme, spýtame sa len na dĺžku vektora `qq`:

```
>> qq=find(relusp>0.05); n=length(qq)
n = 8
```

Vyberieme najlepších 8 (resp. `n`) výsledkov a príslušné názvy strán:

```
>> reluspn=relusp(1:n); poradn=poradie(1:8); Nazvyn=Nazvy(poradn);
>> pie(reluspn,Nazvyn)
```

Pozn.: Ak je súčet hodnôt v prvom argumente menší než 1, Matlab chápe tento vstup ako relatívne počtenosti. Koláč bude teda neúplný.



Histogram

Histogram je špeciálny typ stĺpcového diagramu, ktorý vyjadruje početnosti **tried** definovaných intervalmi na osi x. Šírka všetkých intervalov nemusí byť rovnaká a vtedy triedy nemožno považovať za „rovnocenných súperov“. V záujme spravodlivosti porovnania preto početnosť vyjadruje nie výška, ale plocha stĺpca.

Na písomnej skúške bolo možné získať najviac 300 bodov. Hodnotenie študenta v závislosti od bodov je nasledovné:

0-150	150-200	200-240	240-270	270-290	290-300
F	E	D	C	B	A

Hraničný bod medzi triedami sa považuje za súčasť vyššej triedy. Výnimkou bude pre nás len najvyššia hodnota 300, ktorú ponecháme v triede A. Hoci je naše uvažovanie prirodzené, Matlab takúto výnimku neuznáva a pre hodnotu 300 vytvorí osobitnú triedu. Táto pedantnosť nám trochu skomplikuje postup, ale v určitých situáciách má význam a ešte ju oceníme.

Na skúšku prišlo 721 študentov a ich bodové zisky sú nasledujúce:¹

```
>> sk=ceil(rand(1,721).^2*300);
```

Do vektora ht umiestnime hranice tried:

```
>> ht=[0 150 200 240 270 290 300];
```

Hraničných bodov je 7, tried je 6. Pýtame sa, koľko študentov dostalo jednotlivé známky, čiže aké sú početnosti tried:

```
>> pt=histc(sk,ht)
pt = 489 95 67 33 23 13 1
```

Dostali sme 7 údajov. Posledný hovorí o tom, že jeden človek dosiahol plný počet bodov. Z hľadiska nášho chápania však patrí do triedy A, takže ho tam mechanicky presunieme:

```
>> k=length(pt); ptt=[pt(1:k-2), pt(k-1)+pt(k)]
ptt = 489 95 67 33 23 14
```

Triedy majú rôznu šírku a ich početnosť chceme vyjadriť plochou. Matlab na vykreslenie diagramu však potrebuje výšky stĺpcov (ozn. vs):

```
>> vs=ptt./(ht(2:end)-ht(1:end-1))
vs = 3.2600 1.9000 1.6750 1.1000 1.1500 1.4000
```

Početnosti (plochu) tu delíme po zložkách šírkou tried, čím dostávame výšky stĺpcov.

¹ Získaný vektor sk považujte za **daný**. Vzorec, ktorým sme ho vytvorili, nie je nutné chápať. Vzhľadom na to, že ide o náhodný vektor, môžu čitateľovi pri matlabe vychádzať iné hodnoty ako uvádzame tu.

Poznámky:

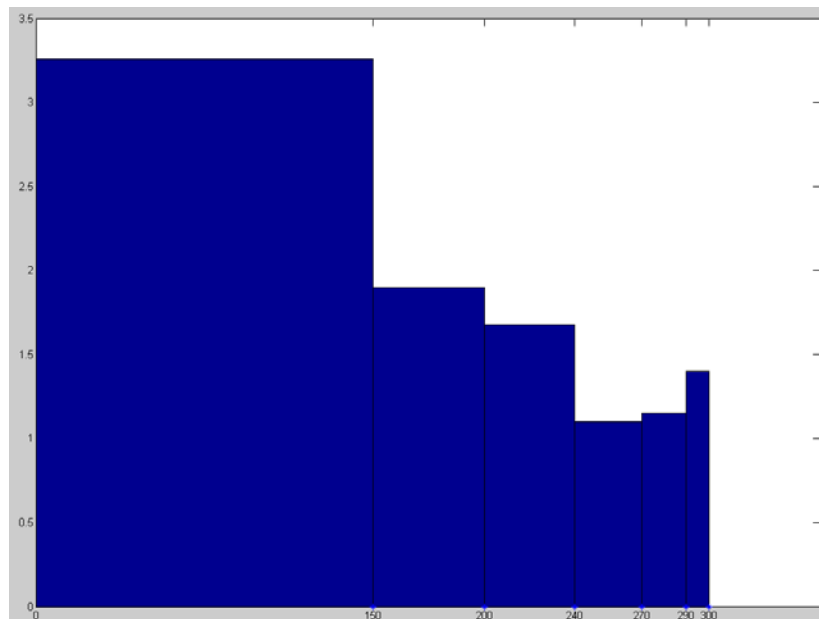
Vo vektore `ht(2:end)-ht(1:end-1)` sú šírky tried (pravé hraničné body mínus ľavé). Parametrom `end` označujeme index poslednej zložky vektora. Matlab tomu rozumie a my nemusíme zvlášť merať dĺžku `ht`, hoci v tomto prípade už vieme, že je to `k`, ktoré sme už vypočítali vyššie.

Vo vzorci sa vyskytuje delenie dvoch vektorov rovnakej dĺžky po zložkách. V Matlabe ho značíme bodkou pred delením (analogicky násobenie či umocňovanie po zložkách zapíšeme bodkou a príslušnou operáciou). Ak máme dva vektory $y=[y_1 \ y_2 \ y_3 \ \dots]$ a $x=[x_1 \ x_2 \ x_3 \ \dots]$, ich podiel po zložkách je $y./x=[y_1/x_1 \ y_2/x_2 \ y_3/x_3 \ \dots]$.

Ak by sme na bodku zabudli, Matlab by delenie y/x realizoval ako maticovú operáciu a našiel by nám číslo a , ktoré v zovšeobecnenom zmysle rieši rovnicu $y=a*x$.

Histogram nakreslíme pomocou príkazu `bar`. Chceme, aby sa stĺpce vykreslili medzi príslušnými hraničnými bodmi – to zariadíme parametrom `'histc'`. Argumentom sú hranice tried a výšky stĺpcov. Ku vektoru `vs` musíme pridať nulu – syntax Matlabu si vyžaduje vektor rovnakej dĺžky ako je vektor hraničných bodov:

```
>> bar(ht, [vs 0], 'histc')
```



Výsledky skúšok sme spracovali ako triedené údaje. Hneď na začiatku sme študentov rozdelili do tried, t.j. pridelili sme im známky a o ich pôvodné body sme sa nezaujímalí. Pri danom množstve študentov a bodovom rozpätí je to nutná daň za prehľadnosť.

Vráťme sa k pôvodným podrobným údajom – k údaju vo vektore sk. Nebudeme bodové zisky triediť, ale ku každej bodovej hodnote (sú iba celočíselné) budeme chcieť priradiť počet študentov, ktorí ju dosiahli. Hľadáme teda vektor dĺžky 301 s počtami študentov, ktorí dosiahli body 0, 1, 2, ... 300.

Táto úloha sa dá riečiť rovnakým spôsobom ako predošlá. Hranice tried si určíme s rozstupom 1 bodu:

```
>> ht1=0:300;
```

Početnosť týchto tried sa rovná počtu študentov s bodmi na ľavej hranici triedy. Oceníme tu, že Matlab hodnotu 300 považuje za samostatnú triedu.

```
>> pt1=histc(sk,ht1);
```

Vektor pt1 je príliš dlhý na vypisovanie. Opýtajme sa však aspoň, koľkí študenti majú 0 bodov a 1 bod:

```
>> body0=pt1(1), body1=pt1(2)
```

```
body0 = 0  
body1 = 31
```

Sú nejaké bodové úrovne, ktoré nedosiahol nikto? Zistíme, na ktorých pozíciách pt1 sa nachádza 0 a vypíšeme si bodové hodnoty týchto pozícií v ht1.

```
>> poz=find(pt1==0); nikto=ht1(poz)
```

```
nikto =
```

```
0 85 89 92 94 95 112 119 123 125 135 140 145 155 163 173 174 177 180 184 185 187 193  
196 205 209 218 219 220 225 226 231 234 241 244 245 246 250 251 258 262 264 268 269 270  
272 275 277 278 280 281 282 284 291 294 297
```

Aký bodový zisk bol najčastejší?

```
>> mapt=max(pt1), pom=find(pt1==mapt); ht1(pom)
```

```
mapt = 31  
ans = 1
```

Najviac bolo 1-bodových, až 31.

Koľko ľudí malo z testov nepárny počet bodov?

```
>> nepar=2:2:300;  
>> sum(pt1(nepar))
```

```
ans = 368
```

Vektor nepar obsahuje indexy pozícií v ht1, kde sa nachádzajú nepárne hodnoty (0 je na 1. pozícií, 1 na druhej, ... 300 na 301-ej).