

Programovanie v prostredí

V. Ďalšie možnosti vizualizácie dát

Aleš Kozubík

Katedra Matematických Metód a Operačnej Analýzy

1.12.2019

Stĺpcový graf

Stĺpcový graf používame na zobrazenie rozdelenia resp. frekvencie výskytu kategoriálnych dát, s využitím zvislých alebo horizontálnych stĺpcov.

Základný príkaz má tvar

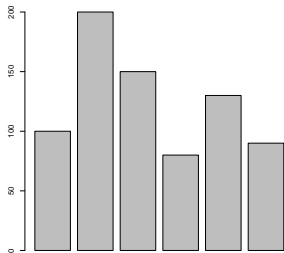
```
barplot(height)
```

kde `height` je vektor alebo matica s údajmi.

Stĺpcový graf - príklad

Jednoduchý stĺpcový graf vytvoríme napr.

```
barplot(c(100, 200, 150, 80, 130, 90))
```



Stĺpcový graf

Ak chceme využiť horizontálne stĺpce v grafe, doplníme voľbu `horiz=T`

Základný príkaz má potom tvar

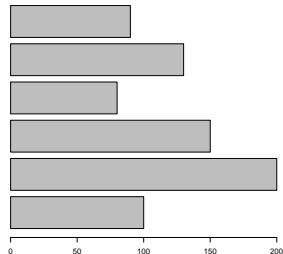
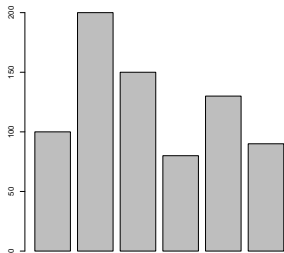
```
barplot(height, horiz=T)
```

kde `height` je vektor alebo matica s údajmi.

Stĺpcové graf - príklad

Jednoduché stĺpcové grafy vytvoríme napr.

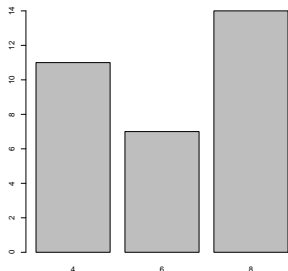
```
barplot(c(100,200,150,80,130,90))  
barplot(c(100,200,150,80,130,90),horiz=T)
```



Stĺpcový graf - pomenovania stĺpcov

Ak využijeme zabudovaný dataset `mtcars`, pripravíme si najskôr vektor s počtami (napríklad podľa počtu valcov motora) a ten následne zobrazíme.

```
> count <- table(mtcars$cyl)
> barplot(count)
```

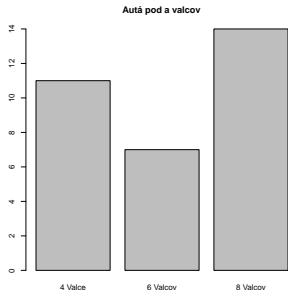


Čo chceme doplniť:

- hlavný nadpis,
- zrozumiteľnejšie popisy stĺpcov

Stĺpcový graf - nadpis a názvy stĺpcov

```
> count <- table(mtcars$cyl)
> barplot(count, main="Autá podľa valcov",
  names.arg=c("4 Valce", "6 Valcov", "8 Valcov"))
```

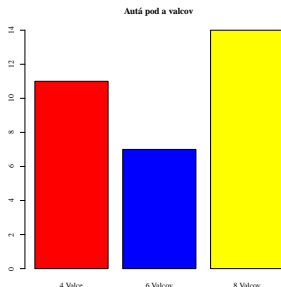


Čo chceme doplniť:

- farebnosť,
- popis osi x

Stĺpcový graf - farebnosť stĺpcov

```
> count <- table(mtcars$cyl)
> barplot(count, main="Autá podla valcov",
names.arg=c("4 Valce", "6 Valcov", "8 Valcov"),
col=c("red", "blue", "yellow"), xlab="Pocet valcov")
```



Stĺpcový graf - viacrozmerne dáta

Využijeme zabudovaný dataset Titanic

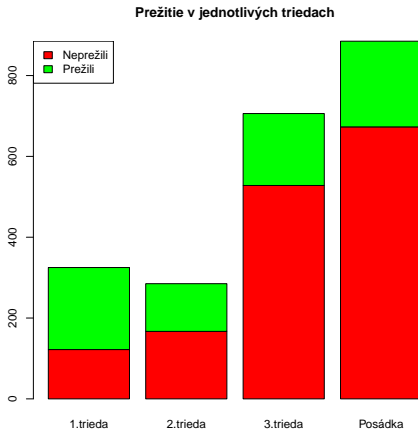
Vygenerujeme si tabuľku dát, ktoré budeme zobrazovať:

```
> titanic.data<-t(margin.table(Titanic,c(1,4)))  
> titanic.data  
      Class  
Survived 1st 2nd 3rd Crew  
      No  122 167 528  673  
      Yes 203 118 178  212
```

Stĺpcový graf - viacrozmerne dáta, kód

```
> barplot(titanic.data,  
+ main = "Prezitie v jednotlivých triedach",  
+ col = c("red", "green"),  
+ names.arg=c("1. trieda", "2. trieda", "3. trieda", "Posadka")  
+ )  
> legend("topleft",  
+ c("Neprezili", "Prezili"),  
+ fill = c("red", "green")  
+ )
```

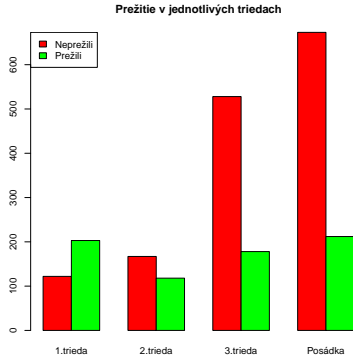
Stĺpcový graf - viacrozmerne dáta, výsledok



Stĺpcový graf - viacrozmerné dáta

Stĺpce vedľa seba

Ako argument funkcie `barplot()` pridáme `beside=TRUE`.



Koláčové grafy

Sú zaužívané v podnikateľskom prostredí a obchode.

V štatistike sú pomerne zahnávané, nakoľko dĺžku vieme posúdiť lepšie než objem. Preto sa viac odporúča používať stĺpcové grafy

To je pravdepodobnou príčinou obmedzených možností koláčových grafov oproti iným softvérom.

Koláčový graf

Základný príkaz pre vytvorenie koláčového grafu je `pie(x,labels)`, kde:

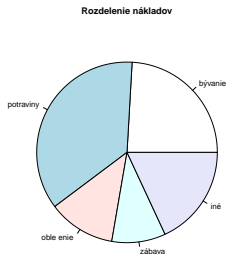
- `x` je vektor hodnôt (počtov) v jednotlivých kategóriách,
- `labels` je vektor s pomenovaniami kategórií.

Pripravíme si údaje pre zobrazenie:

```
> naklady<-c(200,300,100,80,150)
> polozky<-c("byvanie","potraviny","oblecenie",
  "zabava","ine")
```

Koláčový graf - výsledok

```
> pie(naklady, polozky, main="Rozdelenie nakladov")
```

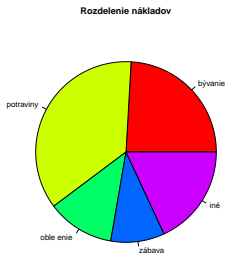


Čo chceme upraviť:

- farebnosť,
- doplniť číselné údaje.

Koláčový graf - zmena farebnosti

```
> pie(naklady, polozky, main="Rozdelenie nakladov",  
      col=rainbow(length(naklady)))
```

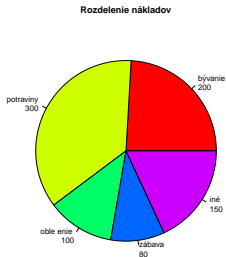


Čo chceme upraviť:

- doplniť číselné údaje.

Koláčový graf - hodnoty v popisoch

```
> popisy <- paste(položky, "\n", naklady, sep="")  
> pie(naklady, popisy, main="Rozdelenie nakladov",  
      col=rainbow(length(naklady)))
```



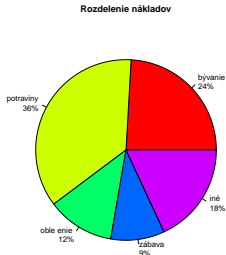
Čo chceme upraviť:

- prepočet na percentá,
- vytvoriť 3D efekt.

Koláčový graf - hodnoty v percentách

Upravíme popisy

```
> popisy <- paste(polozky, "\n",  
  trunc(100*naklady/sum(naklady)), "%", sep="")  
> pie(naklady, popisy, main="Rozdelenie nakladov",  
  col=rainbow(length(naklady)))
```



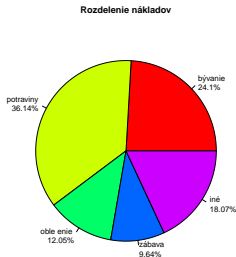
Pri zaokrúhlení:

- súčet nemusí byť 100%,
- spresníme vyšším počtom desatinných miest.

Koláčový graf - hodnoty v percentách

Namiesto `trunc()` použijeme `round()`

```
> popisy <- paste(polozky, "\n",  
  round(100*naklady/sum(naklady), digits=2),  
  "%", sep=" ")
```



Pri zaokrúhlení:

- súčet už je 100%,
- počet desatinných miest je ale vždy limitovaný prehľadnosťou.

Koláčový graf 3D - knižnica

Potrebuje ku práci knižnicu (balíček) "plotrix"

Načítanie knižnice realizujeme príkazom `library(nazov)`, kde `nazov` je pomenovanie požadovanej knižnice (zapísaný v úvodzovkách). V našom prípade

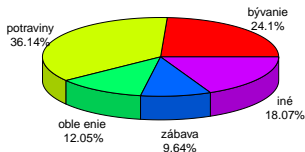
```
> library("plotrix")
```

Pokiaľ balíček nie je nainštalovaný, doinštalujeme ho príkazom `install.packages(nazov)`. Inštalátor ponúkne výber vhodného zrkadla repozitára knižníc a po jeho voľbe balíček nainštaluje.

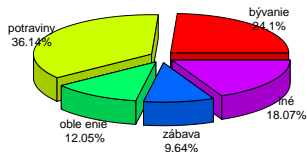
Koláčový graf - 3D efekt

```
> library("plotrix")  
> pie3D(naklady, labels=popisy, main="Rozdelenie nakladov")  
> pie3D(naklady, labels=popisy, explode=0.1,  
main="Rozdelenie nakladov")
```

Rozdelenie nákladov



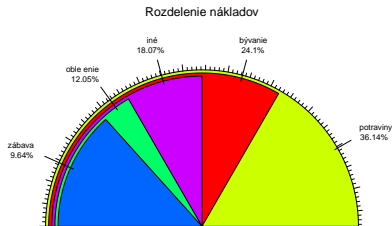
Rozdelenie nákladov



Koláčový graf - vejárové porovnanie

Chceme priamo porovnať veľkosť podielov.

```
> library("plotrix")  
> fan.plot(naklady, labels=popisy,  
  main="Rozdelenie nakladov", max.span=pi,  
  ticks=max(naklady))
```

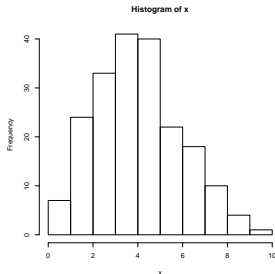


Histogram

Histogram je stĺpcový graf pre grafické zobrazenie počností hodnôt náhodnej premennej resp. zvolených intervalov hodnôt spojitej náhodnej premennej.

Ak x je vektor nameraných hodnôt, tak histogram zobrazíme jednoduchým príkazom `hist(x)`

```
> x<-rbinom(200,15,0.3)  
> hist(x)
```



Histogramy - prispôbenie

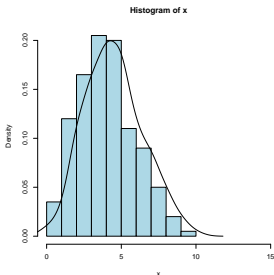
Vzhľad histogramu môžeme meniť pomocou týchto argumentov:

- farebnosť pomocou voľby `col="farba"`,
- zobrazenie relatívnej početnosti namiesto absolútnej voľbou `freq=FALSE`,
- rozsah osi `x` voľbou `xlim=c(min,max)`,
- pridanie funkcie hustoty do grafu.

Histogramy

Vyfarbíme stĺpce, prispôbíme rozsah osi x počtu opakovaní, vyznačíme graf funkcie hustoty.

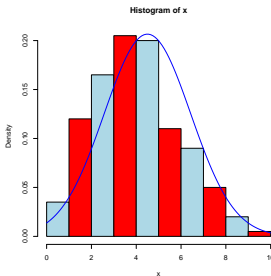
```
> hist(x, xlim=c(0,15),  
  freq=FALSE,  
  col="lightblue")  
> lines(density(x),  
  lwd=2, col="black")
```



Histogramy

Vyfarbíme stĺpce dvomi farbami, vyznačíme graf funkcie hustoty normálneho rozdelenia.

```
> hist(x, freq=FALSE,
      col=c("lightblue", "red"))
> xfit<-seq(min(x), max(x),
           length=100)
> yfit<-dnorm(xfit,
             mean=mean(x), sd=sd(x))
> lines(xfit, yfit,
       col="blue", lwd=2)
```



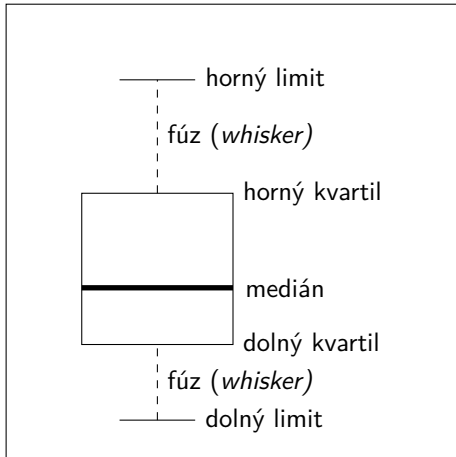
Box plot

Box ploty slúžia na grafickú prezentáciu štatistických dát, zobrazením piatich kvantilových charakteristík:

- medián,
- dolný kvartil,
- horný kvartil,
- minimum,
- maximum.

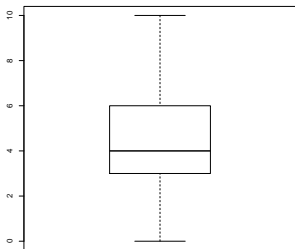
Tiež môže zobrazovať extrémne hodnoty, ktoré sú od mediánu vzdialené viac ako 1.5-násobok medzikvartilového rozpätia.

Box plot



Box plot

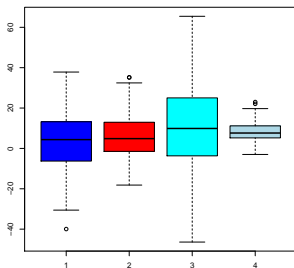
Najjednoduchšie ho zostrojíme pomocou príkazu `\boxplot(x)`, kde `x` je vektor hodnôt, ktorých rozdelenie chceme zobraziť.



Čo chceme upraviť:

- farebnosť,
- porovnať viacero výberov,
- prispôbiť dĺžku fúzov,
- meniť tvar boxov.

Box plot - farebný

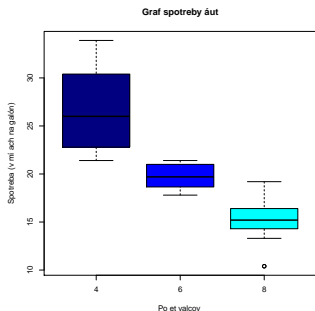


```
> x1<-rnorm(200,3,15)
> x2<-rnorm(200,5,10)
> x3<-rnorm(200,10,20)
> x4<-rnorm(200,8,5)
> boxplot(x1,x2,x3,x4,
  col=c("blue","red",
    "cyan","lightblue"))
```

Box plot - skupiny premenných

Využijeme dataset `mtcars`, z ktorého porovnáme spotrebu áut podľa počtu valcov.

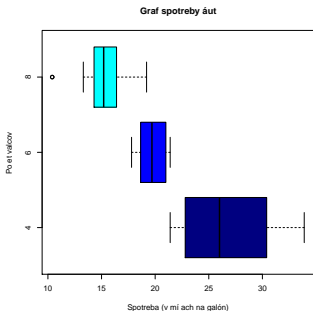
```
> boxplot(mpg~cyl,data=mtcars,  
main="Graf spotreby áut",  
xlab="Pocet valcov",  
ylab="Spotreba (v mil. ach na galón)",  
col=c("navy","blue","cyan"))
```



Box plot - zmena orientácie

Pomocou voľby `horizontal=TRUE`, zmeníme orientáciu boxplotu zo zvislej na vodorovnú.

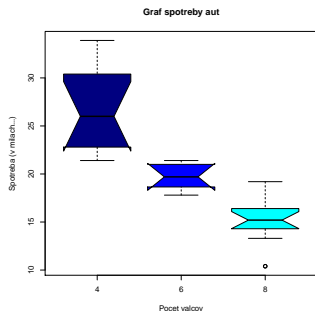
```
> boxplot(mpg~cyl, data=mtcars,
  main="Graf spotreby aut",
  xlab="Pocet valcov",
  ylab="Spotreba (v milach...)",
  col=c("navy", "blue", "cyan"),
  horizontal=TRUE)
```



Box plot - so zárezom

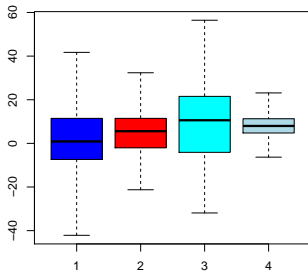
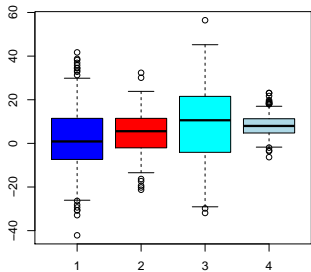
Pomocou voľby `notch=TRUE`, zmeníme tvar boxu. Ak sa zárezy neprekrývajú, je to silným prejavom rozdielnosti mediánov.

```
> boxplot(mpg~cyl, data=mtcars,
  main="Graf spotreby aut",
  xlab="Pocet valcov",
  ylab="Spotreba (v milach...)",
  col=c("navy", "blue", "cyan"),
  notch=TRUE)
```



Box plot - dĺžka fúzov

Dĺžku fúzov regulujeme parametrom `range=<c>`, pričom hodnota `<c>` udáva násobok medzikvartilového rozpätia, ktorým je určená dĺžka fúzov. Implicitná hodnota je 1.5.



Q-Q plot

Q-Q plot (kvantil – kvantil) je grafom kvantilov jedného rozdelenia (empirického) oproti kvantilom iného rozdelenia (teoretického). Je to vlastne parametrická krivka v rovine, pričom jej súradnice sú dané kvantilmi jednotlivých rozdelení.

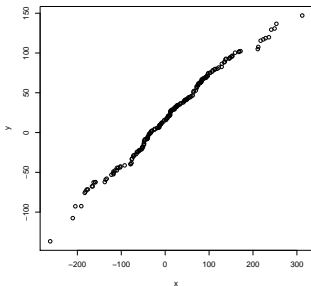
Interpretácia:

- Ak sú rozdelenia identické, graf „kopíruje“ priamku $x = y$.
- Ak sa rozdelenia zhodujú po lineárnej transformácii, krivka má tvar nejakej priamky, nie nutne $x = y$.
- Zakrivenie indikuje zošikmenie jedného z rozdelení resp. že jedno z rozdelení má ťažšie chvosty.

Q-Q plot

Pre konštrukciu Q-Q plotu je kdispozícii príkaz `qqplot()`.
Základný tvar je `qqplot(x,y)`, kde `x` a `y` sú dva náhodné výbery,
chceme porovnať, či pochádzajú z toho istého typu rozdelenia.

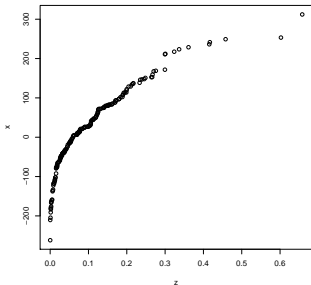
```
> x<-rnorm(200,10,100)
> y<-rnorm(200,20,50)
> qqplot(x,y)
```



Q-Q plot

Porovnanie náhodných výberov z normálneho a exponenciálneho rozdelenia pomocou q-q plotu:

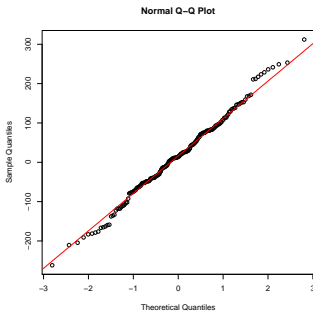
```
> z<-rexp(200,10)  
> qqplot(z,x)
```



Q-Q plot porovnanie s normálnym rozdelením

Pre priame porovnanie výberu s normálnym rozdelením je k dispozícii funkcia `qqnorm()`. Funkcia `qqline()` zobrazuje priamku spájajúcu prvý a tretí kvartil.

```
> qqnorm(x)
> qqline(x, col="red")
```



Q-Q plot porovnanie s normálnym rozdelením

Ak pôvodné rozdelenie nie je normálne.

```
> qqnorm(z)  
> qqline(z, col="red")
```

Exponenciálne rozdelenie má
ťažšie chvosty než normálne.

