

Programovanie v prostredí

II. Začíname s grafikou

Aleš Kozubík

Katedra Matematických Metód a Operačnej Analýzy

25.10.2019

Čo si ukážeme

Ako základ práce pri vytváraní grafov si ukážeme:

- vytváranie a ukladanie grafov,
- užívateľské úpravy symbolov, čiar, farieb a osí,
- textové popisky a nadpisy,
- riadenie rozmerov grafu,
- spájanie viacerých grafov do jedného obrázku.

Jednoduchý graf

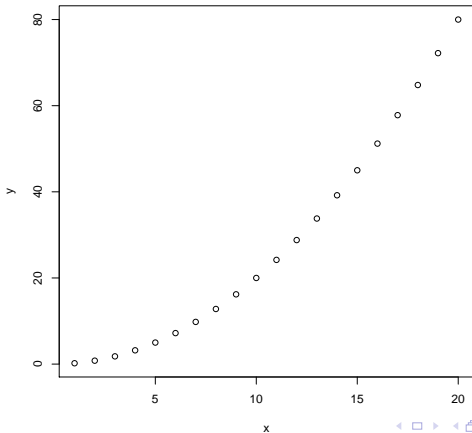
Základom pre vykreslenie grafu je funkcia `plot()`.

Jej základnými argumentmi sú hodnoty `x` a `y`, ktoré majú byť vykreslené.

Príklad

```
> x<-1:20  
> y<-x^2/5  
> plot(x,y)
```

Jednoduchý graf – výsledok



Jednoduchý graf – uloženie do súboru

V prvom rade sa treba rozhodnúť o formáte ukladaného obrázku. Ak napr. chceme obrázok vo formáte `.pdf`, použijeme funkciu `pdf()`, ktorej základným parametrom je názov súboru (ev. aj s celou cestou).

```
> pdf("mojprvyplot.pdf")  
> plot(x,y)  
> dev.off()  
>
```

Prípustné formáty ukladania grafov

Okrem už odskúšaného `.pdf` formátu môžeme ukladať ešte:

Funkcia	Formát
<code>pdf()</code>	vektorový formát <code>.pdf</code>
<code>svg()</code>	vektorový formát <code>.svg</code>
<code>postscript()</code>	vektorový formát <code>.ps</code>
<code>png()</code>	formát <code>.png</code> s vysokým rozlíšením
<code>jpeg()</code>	formát <code>.jpg</code> s vysokým rozlíšením
<code>bmp()</code>	formát <code>.bmp</code> s vysokým rozlíšením
<code>tiff()</code>	formát <code>.tiff</code> s vysokým rozlíšením

Vybrané parametre funkcií pre vektorovú grafiku

Parameter	Význam
filename	názov súboru (s celou cestou, ak treba)
width	šírka v palcoch (default hodnota 7)
height	výška v palcoch (default hodnota 7)
bg	farba pozadia (default hodnota transparent)
fg	farba popredia (default hodnota black, iba pdf())
family	názov počiatočnej skupiny fontov (default "Helvetica") alternatívne "Times", "Courier"

Vybrané parametre funkcií pre bitmapy

Parameter	Význam
filename	názov súboru (s celou cestou, ak treba)
width	šírka v palcoch (default v pixeloch, hodnota 480)
height	výška v palcoch (default v pixeloch, hodnota 480)
bg	farba pozadia (default hodnota "white")
res	rozlíšenie (default hodnota 72 dpi)
units	jednotky merania (default px pixely) alternatívy cm, mm alebo in (palce)

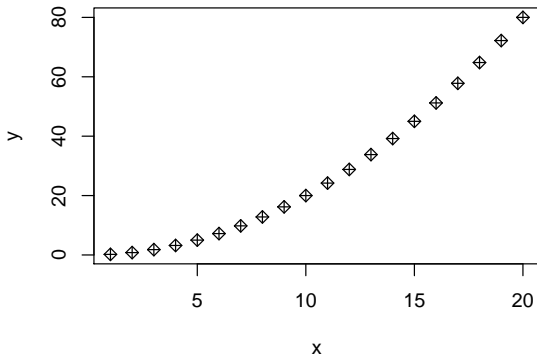
Symoly pre zobrazenie bodov

Vzhľad bodov sa riadi hodnotou parametra `pch` funkcie `plot()`.

Bod	pch=	Bod	pch=	Bod	pch=	Bod	pch=	Bod	pch=
□	0	○	1	△	2	+	3	×	4
◇	5	▽	6	⊠	7	*	8	⊞	9
⊕	10	☆	11	⊞	12	⊗	13	⊠	14
■	15	●	16	▲	17	◆	18	●	19
●	20	○	21	□	22	◇	23	△	24

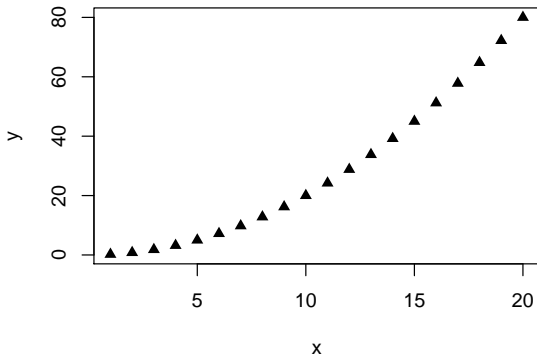
Symbols pre zobrazenie bodov – ukážka

```
plot(x,y,pch=9)
```



Symbols pre zobrazenie bodov – ukážka

```
plot(x,y,pch=17)
```



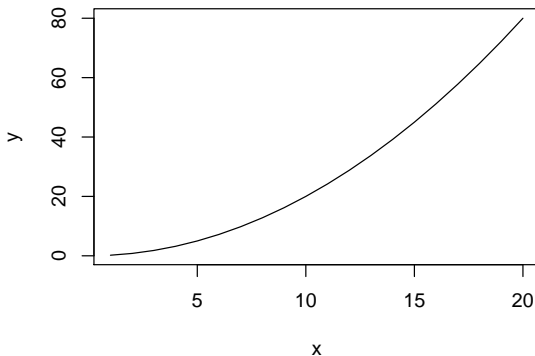
Úprava typu grafu

Typ grafu určuje argument `type` funkcie `plot`.

Typ	Význam
p	bodový graf (points), default hodnota
l	spojitý graf (lines)
b	spojitý graf s vyznačenými bodmi (both)
c	ako b ale značky bodov sú vynechané
o	ako b ale značky bodov sú prekreslené (overplotted)
h	hodnoty vykreslené ako histogram
s	schodovitý graf (stairs)

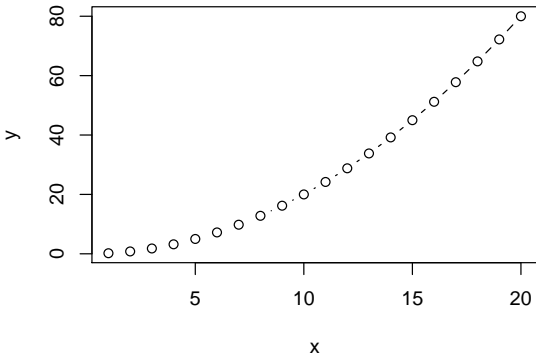
Úprava typu grafu – ukážky

```
plot(x,y,type="l")
```



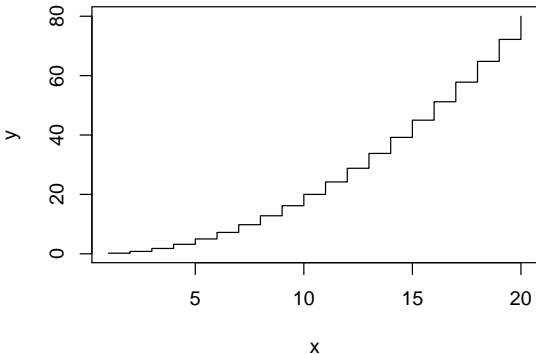
Úprava typu grafu – ukážky

```
plot(x,y,type="b")
```



Úprava typu grafu – ukážky

```
plot(x,y,type="s")
```



Druh čiary pre zobrazenie

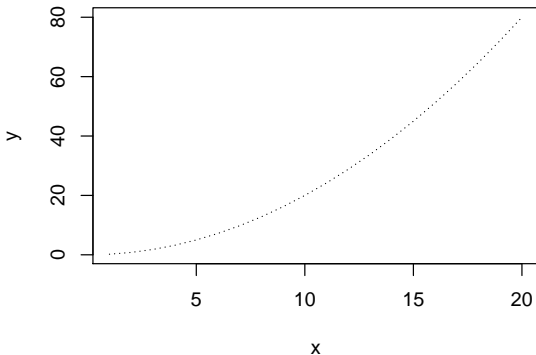
Upravuje sa hodnotou parametra `lty` funkcie `plot()`.

Hodnota	Význam
1	súvislá (plná) čiara, default hodnota
2	čiarkovaná čiara
3	bodkovaná čiara
4	striedavo bodky – čiarky
5	čiarkovane s dlhšími čiarkami
6	bodkočiarkovane s dlhšími čiarkami

Hrúbku čiary riadi hodnota parametra `lwd`.

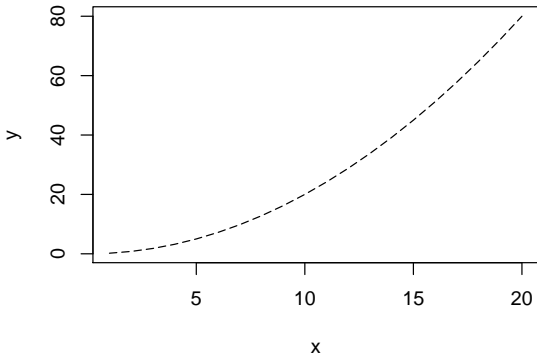
Úprava typu čiary – ukážky

```
plot(x,y,type="l",lty=3)
```



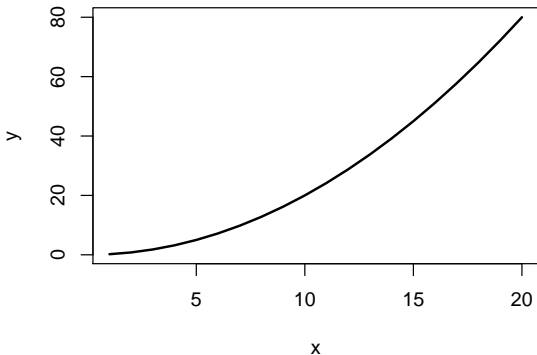
Úprava typu grafu – ukážky

```
plot(x,y,type="l",lty=5)
```



Úprava typu grafu – ukážky

```
plot(x,y,type="l",lty=1,lwd=2)
```



Riadenie farebnosti grafu

Upravuje sa hodnotou parametra `col` funkcie `plot()`.

Konkrétnu farbu je možné zadať tromi spôsobmi:

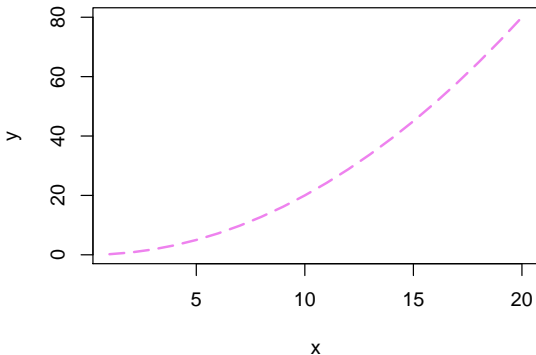
- pomenovaním farby (typ `character`),
- číslom farby, **Moja skúsenosť – číslo nekorešponduje so vzorkovníkom**
- RGB kódom farby (typ `character`)

Zoznam pomenovaní farieb získame pomocou funkcie `colors()`.

Pozn. 1: Niektoré funkcie pre grafy akceptujú ako farbu vektor hodnôt, ktoré sa potom cyklicky opakujú.

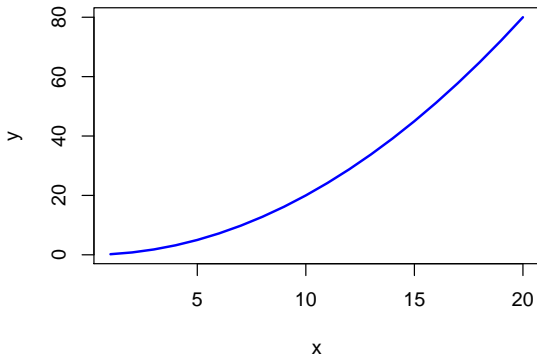
Úprava farby grafu – ukážky

```
plot(x,y,lty=5,type="l",col="violet",lwd=2)
```



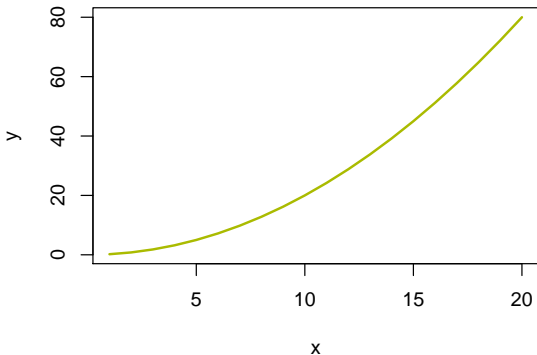
Úprava farby grafu – ukážky

```
plot(x,y,lty=1,type="l",col=636,lwd=2)
```



Úprava typu grafu – ukážky

```
plot(x,y,lty=1,type="l",col="#AABB00",lwd=2)
```



Ďalšie nastavenia farieb

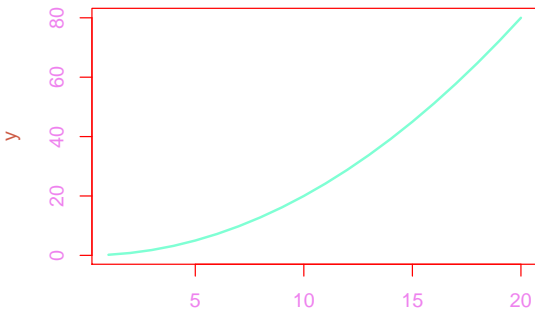
Okrem samotného grafu je možné regulovať farebnosť ostatných prvkov:

Parameter	Význam
col.axis	Farba značiek na jednotlivých osiach
col.lab	Farba popisov osí
col.main	Farba hlavného nadpisu
col.sub	Farba podnadpisu
bg	Farba pozadia grafu
fg	Farba popredia grafu (osi, boxy a pod.)

Ďalšie nastavenia farieb – ukážka

```
plot(x,y,lty=1,type="l",col="aquamarine",lwd=2,  
col.axis="violet",col.main="green",main="Nadpis",  
fg="red",col.lab="coral3")
```

Nadpis



Nadpisy a podnadpisy

Nadpis a podnadpis do hotového grafu pridávame pomocou funkcie `title()`.

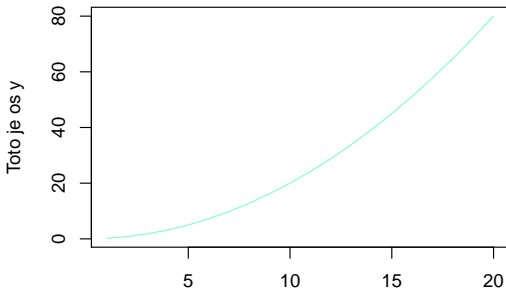
Ak chceme pridávať popisy dodatočne, je treba na to myslieť už pri vytváraní grafu a zadať funkcii `plot()` parametre `xlab=""` a `ylab=""`, inak sa budú texty prepisovať.

Parameter	Význam
<code>main</code>	Hlavný nadpis
<code>sub</code>	Podnadpis
<code>xlab</code>	Popis osi x
<code>ylab</code>	Popis osi y

Nadpisy a podnadpisy – ukážka

```
plot(x,y,lty=1,type="l",col="aquamarine",xlab="",ylab="")  
title(main="Hlavný nadpis",sub="Podnadpis",  
xlab="Toto je os x",ylab="Toto je os y")
```

Hlavný nadpis



Toto je os x
Podnadpis

Textové popisy

Ľubovoľný text je možné do obrázka pridať pomocou funkcií `text()` a `mtext()`.

Funkcia `text()` slúži na pridávanie textu do plochy obrázku.

Funkcia `mtext()` umiestňuje text na niektorý zo štyroch okrajov.

Argumenty funkcie `text()`

```
text(location, "Text", pos=n, ...)
```

Argument	Význam
<code>location</code>	súradnice x a y pozície, kam sa umiestni text
<code>pos</code>	pozícia vzhľadom na aktuálnu plohu 1=bottom, 2=left, 3=above, 4=right

Text je alternatívne možné umiestniť pomocou myši, ak zadáme pozíciu ako funkciu `locator(1)`.

Tiež je možné zadávať ďalšie bežné argumenty ako `col`, `font`, `cex` a pod.

Argumenty funkcie `mtext()`

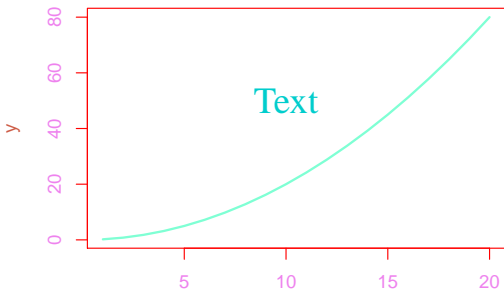
```
mtext("Text", side, line=n, ...)
```

Argument	Význam
<code>side</code>	určuje stranu, na ktorý okraj sa má text umiestniť 1=bottom, 2=left, 3=top, 4=right
<code>line</code>	číslo riadku na danom okraji, začína od 0

Vlastný vložený text – ukážka

```
plot(x,y,lty=1,type="l",col="aquamarine",  
lwd=2,col.axis="violet",col.main="green",  
main="Nadpis",fg="red",col.lab="coral3")  
text(10,50,"Text",col="cyan3",family="Times",cex=2)
```

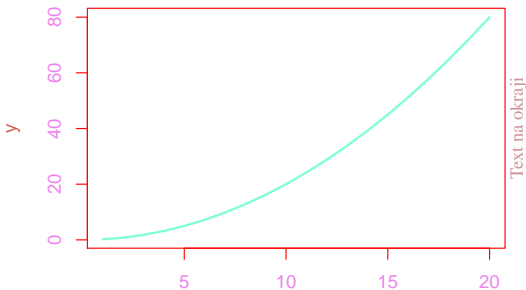
Nadpis



Vlastný text na okraji – ukážka

```
plot(x,y,lty=1,type="l",col="aquamarine",  
lwd=2,col.axis="violet",col.main="green",  
main="Nadpis",fg="red",col.lab="coral3")  
mtext("Text...",side=4,line=0,col="pink3",family="Times")
```

Nadpis



Vložený text – pokročilejšia ukážka

Funkcia `plot()` pracuje s dvojicou vektorov, ako základnými argumentmi pre zobrazenie.

Ak disponujeme vektorom, ktorý obsahuje pomenovania zobrazených údajov, môžeme tieto mená automaticky priradiť na príslušné pozície.

Stačí, ak vektor s požadovanými textami zadáme ako `text` pre zobrazenie v argumente funkcie `text()`.

Ilustrujeme si to na zabudovanej databáze `mtcars`

Aby sme nemuseli vypisovať dlhé názvy, databázu pripojíme ku pracovnej ploche príkazom `attach(mtcars)`.

Vložený text – pokročilejšia ukážka

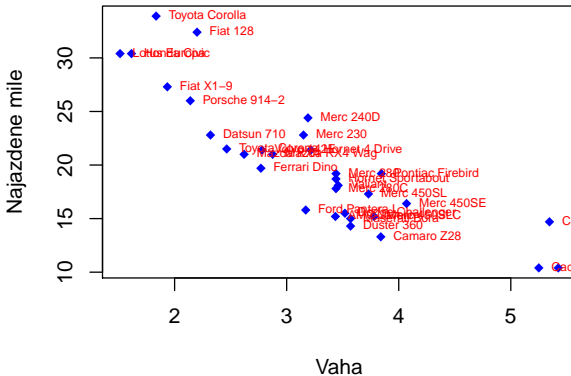
Obrázok (graf) vytvoríme pomocou dvojice príkazov:

```
> plot(wt, mpg, main="Najazdene_mile_vs._vaha_auta",  
+      xlab="Vaha", ylab="Najazdene_mile",  
+      pch=18, col="blue")  
> text(wt, mpg, row.names(mtcars),  
+      cex=0.6, pos=4, col="red")
```

Databázu následne odpojíme od pracovnej plochy príkazom `detach(mtcars)`.

Výsledok

Najazdene mile vs. hmotnosť auta



Časti textu mimo plochu obrázka

Niekedy sa stáva, že časť textu presiahne rozmery plochy obrázka a text sa tak nezobrazí celý.

Toto správanie je možné potlačiť nastavením argumentu `xpd` funkcie `text()` na hodnotu `TRUE`.

Vlastné súradnicové osi

Užívateľ si môže zobrazit' vlastné súradnicové osi pomocou funkcie `axis()`.

Ak sa rozhodneme zobrazovať vlastné súradnicové osi, je potrebné vo funkcii `plot()` použiť voľbu `axes=FALSE`, ktorá potláča zobrazenie oboch osí.

Podobne voľby `xaxt="n"` resp. `yaxt="n"` potláčajú vykreslenie osi `x` resp. osi `y`.

```
axis(side , at= , labels= , pos= , lty= , col= , las= , tck= , ...)+
```

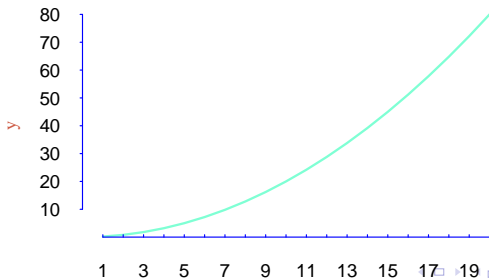
Argumenty funkcie `axis()`

Argument	Význam
<code>side</code>	určuje stranu, na ktorý sa má os umiestniť 1=bottom, 2=left, 3=top, 4=right
<code>at</code>	číselný vektor, určuje pozície značiek na osi
<code>labels</code>	znakový vektor, určuje popis značiek na osi ak je NULL, použijú sa hodnoty z <code>at</code>
<code>pos</code>	určuje pozíciu (hodnotu) v ktorej pretína druhú os
<code>lty</code>	typ čiary pre vykreslenie osi
<code>col</code>	farba čiary pre vykreslenie osi a značiek
<code>las</code>	popisky sú rovnobežne (=0) alebo kolmo (=2) na os
<code>tck</code>	dĺžka značiek, pre vyznačenie hodnôt na osi udáva sa ako pomerná časť veľkosti grafu

Vlastné súradnicové osi – ukážka

```
plot(x,y,lty=1,type="l",col="aquamarine",...)  
axis(side=1,at=x,pos=0,col="blue",  
las=0,tck=0.01,cex=0.7)  
axis(side=2,at=c(10,20,30,40,50,60,70,80),  
pos=0,col="blue",las=2,tck=-0.01)
```

Vlastne osi



Rozdelenie na niekoľko obrázkov

Kombinovanie viacerých grafov do jedného obrázku umožňuje funkcia `par()`

S jej pomocou je možné aj komplexne nastavovať grafické parametre.

Rozdelenie plochy na niekoľko obrázkov docielime vhodnou voľbou argumentu funkcie v tvare `mfrow=c(nrows, ncols)`

Prvá hodnota udáva počet riadkov a druhá počet stĺpcov. Poradie nemožno meniť.

Rozdelenie na niekoľko obrázkov

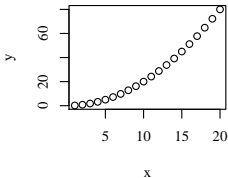
Ukážka – kód

```
x<-1:20
y<-x^2/5
z<-sin(x)
par(mfrow=c(2,2),family="Times")
plot(x,y, main="Prvy")
plot(x,z, main="Druhy")
plot(x,y, main="Prvy",type="l", col="red")
plot(x,z, type="l",main="Druhy",col="aquamarine")
```

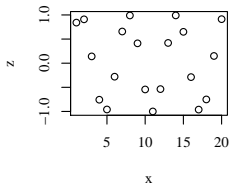
Rozdelenie na niekoľko obrázkov

Ukážka – obrázok

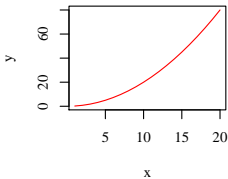
Prvy



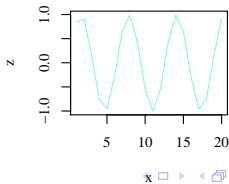
Druhy



Prvy



Druhy



Nastavenie parametrov grafu

Hromadne je možné nastaviť parametre pre graf pomocou funkcie `par()`.

Umožňuje nastaviť prakticky všetky parametre obrázka, napr.

- `family` pre názov použitej rodiny písma,
- `mar` numerický vektor pre určenie veľkosti okrajov, zadáva sa ako `c(dole, vľavo, hore, vpravo)`, čísla udávajú počet riadkov pre príslušný okraj,
- `mfrow, mfcol` dvojjložkový vektor `c(nr, ns)`, nasledujúce obrázky budú vykreslené v schéme `nr` riadkov a `ns` stĺpcov. zapíňat sa budú po riadkoch resp. pri `mfcol` po stĺpcoch.

Nastavenie veľkosti kresliacej plochy

Pre reguláciu veľkosti obrázkov je potrebné si najskôr nastaviť veľkosť zobrazovacieho zariadenia pomocou funkcie `dev.new()`.

Ako argumenty uvidíme `width=<cislo>`, `height=<cislo>`, kde `<cislo>` udáva rozmer v palcoch.

Následne môžeme pomocou funkcie `par()` nastaviť hodnoty parametra `pin` určujúceho rozmery samotnej kresliacej plochy v podobe vektora `c(sirka, vyska)`, opäť rozmery zadávame v palcoch.

Nastavenie veľkosti kresliacej plochy - pokus

Vyskúšajme napríklad

```
> dev.new(width=10,height=6)
> par(pin=c(8,4))
> plot(wt, mpg, main="Najazdene_mile_vs._vaha_auta",
+ xlab="Vaha", ylab="Najazdene_mile",
+ pch=18, col="blue")
> text(wt, mpg, row.names(mtcars), cex=0.6,
+ pos=4, col="red", xpd=TRUE)
>
```

Uloženie hotového obrázku do súboru

Ak sme spokojní so vzhľadom obrázka zobrazeného v grafickom okne, môžeme ho uložiť do súboru.

Používame na to funkciu `dev.print()`, ktorá môže mať tieto argumenty

- `device`= typ zariadenia (formát výstupu), napr. `png`,
- `width`= veľkosť v pixeloch,
- `súbor` pre názov súboru, do ktorého je výstup uložený.

Viaceré grafické okná

Nové grafické zariadenie otvoríme pomocou funkcie `dev.new()`

Grafický výstup sa zobrazuje do aktívneho zariadenia.

Ktoré zariadenie je momentálne aktívne zistíme pomocou funkcie `dev.cur()`.

Zmenu aktívneho zariadenia dosiahneme pomocou funkcie `dev.set(<integer>)`, kde ako argument udávame číslo zariadenia, ktoré chceme aktivovať.

Viacero kriviek v jednom grafe

Prvú krivku v grafe vykreslíme pomocou funkcie `plot()`.

Ďalšie krivky do obrázku pridáme pomocou funkcií `lines()` alebo `points()`, ich syntax je podobná funkcii `plot()`.

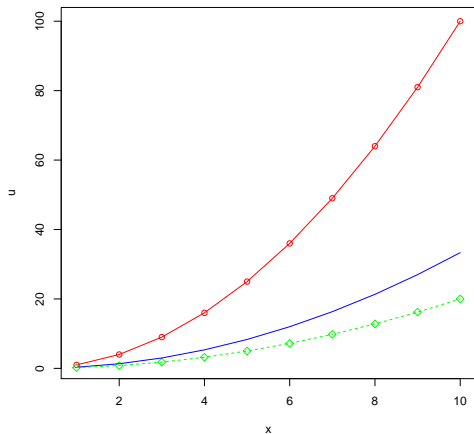
`points()` vykreslí množinu bodov podľa daných súradníc.

`lines()` vykreslí krivku, ktorá vznikne spojením zadaných bodov čiarami.

Viacero kriviek v jednom grafe – príklad

```
> x<-1:10
> y<-x^2/5
> z<-x^2/3
> u<-x^2
> plot(x,u,type="o",col="red")
> points(x,y,pch=5,col="green")
> lines(x,z,pch=4,col="blue")
> lines(x,y,lty=2,col="green")
```

Výsledok



Pridávanie legendy

Ku grafu s viacerými čiarami pridáme legendu pomocou funkcie `legend()`.

Argumenty funkcie `legend()`:

- `x, y` poloha v ploche grafu, zadaná súradnicami v grafe,
- `legend` vektor reťazcov pre popis legendy,
- `col` vektor farieb použitých v grafe,
- `pch` vektor grafových značiek použitých v grafe,
- `lty` vektor udávajúci použité typy čiar,
- `ncol` počet stĺpcov v legende, default je jeden stĺpec.

Ukážka – kód

```
> x <-1:10
> y <-x^2/5
> z <-x^2/3
> u <-x^2
> plot (x,u, type ="o",pch=5,col ="red")
> points (x,y,pch =5, col ="_green_")
> points (x,z,pch=4,col="blue")
> lines (x,y,lty =2, col="_green_")
> lines (x,z,pch =4, col="_blue_")
> legend(1,100,legend=c("y","z","u"),
+ col=c("green","blue","red"),
+ pch=c(2,4,5),lty=c(2,1,1))
>
```

Ukážka – výsledok

