

Zpravodaj Československého sdružení uživatelů TeXu

Marek Pomp

Dobře dokumentované statistické výpočty

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 26 (2016), No. 1-4, 62–77

Persistent URL: <http://dml.cz/dmlcz/150244>

Terms of use:

© Československé sdružení uživatelů TeXu, 2016

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

Článek popisuje implementaci metody dobře dokumentovaných programů v prostředí pro statistické výpočty. Speciálně se věnuje balíčku **Sweave** určeném pro psaní dynamických dokumentů, jejichž výpočty se provádějí za pomoci statistického programu R.

Klíčová slova: literární programování, reprodukovatelný výzkum, Sweave, jazyk R.

Úvod

V roce 1981 Donald E. Knuth formuloval svou ideu literárního programování, na jejímž principu bylo vytvořeno mnoho nástrojů usnadňujících vytváření dobře dokumentovaných programů. Tento trend se dotknul i systémů pro statistické výpočty, viz např. (GANDRUD, 2013).

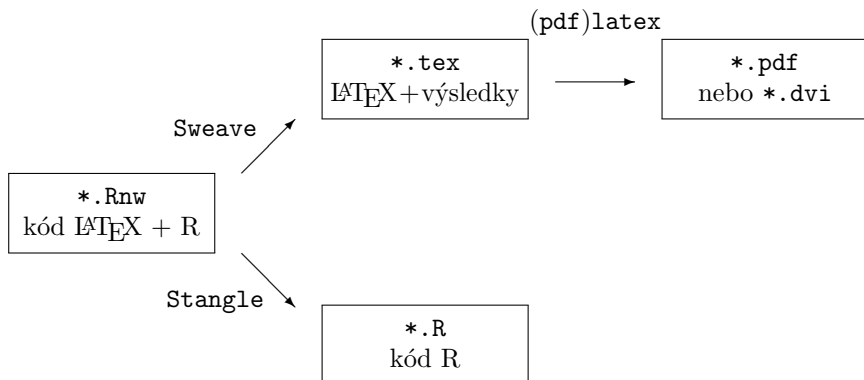
V posledních letech začali statistikové v široké míře používat statistický program R, volně šiřitelnou implementaci jazyka S. V roce 2002 Friedrich Leisch vytvořil systém **Sweave**, který umožňuje kombinovat \LaTeX ovský text se zdrojovým kódem, který je vyhodnocen programem R, viz (LEISCH, 2002) a (LEISCH, 2008).

V roce 2012 navázal na balíček **Sweave** Yihui Xie svým programem **knitr**, viz (XIE, 2013), ve kterém obohatil základní možnosti **Sweave** a integroval výpočty programu R také do dalších značkovacích jazyků, jako je HTML nebo Markdown.

Sweave, principy

Základem při práci s balíkem **Sweave** je kombinace \LaTeX ovského zdrojového souboru s výpočty a procedurami programovacího jazyka R, podle pravidel **noweb** (viz RAMSEY, 1994). Směs textu pro \LaTeX a zdrojového kódu pro R je obvykle uložena v souboru s příponou **Rnw**. Zdrojový soubor **Rnw** se nejprve předloží programu R. R jako preprocesor může mít dvě funkce. Buďto vyhodnotí všechny výpočty a požadované výsledky převede na sekvence, kterým rozumí \TeX . Výsledek práce programu R je čistý soubor určený pro zpracování \LaTeX em, tomuto procesu říkáme *weave*. Druhou variantou je, že ze souboru **Rnw** je extrahován jen zdrojový text pro program R, tzv. proces *tangle*¹ (viz obr. 1).

¹Proces *tangle* spouští funkce **Stangle**, která ze souboru **Rnw** vybírá jen kód určený programu R a ukládá jej do souboru s příponou R.



Obrázek 1: Schéma práce se Sweave a Stangle

Kompilace souboru `*.Rnw` (viz obr. 2) v programu R provádí funkce `Sweave`, která do souboru s příponou `tex` beze změny opíše všechny kód určený pro `TEX`. Kód určený pro program R nechá vyhodnotit a zapíše do stejného souboru výsledky tohoto vyhodnocení spolu s dalšími `TEX`ovými formátovacími příkazy.²

```

> Sweave("prvni.Rnw")
Writing to file prvni.tex
Processing code chunks with options ...
1 : echo keep.source term verbatim (prvni.Rnw:5)
  
```

You can now run `(pdf)latex` on `'prvni.tex'`

Výsledný soubor (na obr. 3) je možné běžným způsobem zpracovat pomocí `LATEXu`. Prostředí `Schunk`, `Sinput` a `Soutput`, která formátují výstupy, jsou definována v balíku `Sweave.sty`, který je součástí běžných distribucí `LATEXu`.

Funkci `Sweave` a následně `LATEX` můžeme spouštět odděleně, `Sweave` v prostředí R a následně `TEX` v operačním systému. Vhodnější ovšem je oba kroky spojit dohromady. Jedna z možností, je spouštět obojí na příkazovém řádku v operačním systému, např. v Linuxu³ pomocí příkazů

```

$ Rscript -e "Sweave('prvni.Rnw');" && pdflatex prvni
nebo v ekvivalentní formě
  
```

```

$ R CMD Sweave prvni.Rnw; pdflatex prvni
  
```

Pomocí předchozích příkazů lze vytvářet složitější skripty a Makefile soubory.

²Příkazovou řádku programu R značíme promptem `>`.

³Příkazovou řádku operačního systému značíme promptem `$`.

```

\documentclass{article}
\usepackage[utf8]{inputenc}
\begin{document}
Výpočet aritmetického průměru z posloupnosti čísel.
<<echo=TRUE, results=verbatim>>=
x <- c(1,2,3,4)
mean(x)
@
\end{document}

```

Obrázek 2: Zdrojový soubor `prvni.Rnw`, kód pro \LaTeX opatřený blokem s kódem pro program R

```

\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{Sweave}
\begin{document}
Výpočet aritmetického průměru z posloupnosti čísel.
\begin{Schunk}
\begin{Sinput}
> x <- c(1,2,3,4)
> mean(x)
\end{Sinput}
\begin{Soutput}
[1] 2.5
\end{Soutput}
\end{Schunk}
\end{document}

```

Obrázek 3: Výsledek práce preprocesoru `Sweave`, soubor `prvni.tex`

```

Výpočet aritmetického průměru z posloupnosti čísel.

x <- c(1,2,3,4)
mean(x)
[1] 2.5

```

Obrázek 4: Výsledek zpracování souboru `prvni.tex` pomocí \LaTeX u, výstup v DVI nebo PDF

Další možností je spouštět `Sweave` i \LaTeX z příkazové řádky programu `R`. Ke spuštění `TEX`u slouží utilitka `texi2dvi` z balíku `tools`, která spouští \LaTeX a `BIBTEX` až do správného vyhodnocení všech křížových odkazů.

```
> Sweave("prvni.Rnw")
> tools::texi2dvi("prvni.tex", pdf=TRUE)
```

V neposlední řadě je možné využít k editaci a následné kompilaci některé z komplexních prostředí pro editaci `TEX`tu. Styl pro `Sweave` je podporován ve většině textových editorů jako je např. Emacs, vim, WindEdt nebo `TEXnicCenter`. Samozřejmostí je podpora pro `LyX`. Za zmínku stojí také `R-Studio` (viz `RSTUDIO TEAM`, 2015), propracovaný nástroj primárně určený pro práci s programem `R` a vytváření dokumentů obsahujících kód jazyka `R`.

U souborů s jiným kódováním než `ASCII` rozpozná `Sweave` typ kódování z \LaTeX ovského `\usepackage[...]{inputenc}`, ale je možné kódování vnutit funkci `Sweave` parametrem `encoding`, např.

```
$ Rscript -e "Sweave('prvni.Rnw', encoding='utf8')"
```

Blok kódu a jeho základní parametry

Do \LaTeX ovského zdroje se části určené programu `R` vkládají jako tzv. *chunks* — *bloky kódu*. Podle zvyklostí v `noweb` je blok kódu na začátku omezen řádkem obsahujícím posloupnost znaků `<<parameter>>=` a na konci řádkem s jediným znakem `@`. Parametry v hlavičce bloku určují jeho další chování, obvykle formátování výstupu. Příklad zdrojového souboru s jedním blokem kódu pro `R` je na obr. 2. Tento blok má dva parametry `echo=TRUE` a `results=verbatim`. Oba parametry jsou v tomto příkladu nadbytečné, protože uvedené hodnoty parametrů jsou ve `Sweave` přednastaveny a chování bloku by bylo stejné, jako kdyby byla hlavička prázdná `<<>>=`.

Pro ortodoxní \LaTeX isty je možné blok `R`-kovského kódu uzavřít do prostředí `Scode`.

```
\begin{Scode}{parameter}
kód pro R
\end{Scode}
```

Aby `Sweave` správně rozpoznal syntaxi pro zápis bloků, měl by soubor s bloky vymezenými prostředím `Scode` mít příponu `Rtex`. Typ zápisu pro bloky lze měnit průběžně v jednom souboru. Změnu syntaxe oznámíme příkazem

```
\SweaveSyntax{SweaveSyntaxLatex}
nebo
\SweaveSyntax{SweaveSyntaxNoweb}
```

Popišme ty nejužívanější parametry pro blok R-kovského kódu. (Hodnoty parametrů jsou uvedeny v závorce, první z uvedených hodnot je ve **Sweave** standardně přednastavena. V příkladech je vlevo blok, tak jak je zapsaný v souboru ***.Rnw** a vpravo výstup ve finálním ***.pdf** souboru.)

echo (TRUE/FALSE) — ukáže/skryje zdrojový kód pro R.

```
<<>>= > x <- c(1,2,3,4)
x <- c(1,2,3,4) > mean(x)
mean(x) [1] 2.5
@
<<echo=FALSE>>= [1] 2.5
x <- c(1,2,3,4)
mean(x)
@
```

results (verbatim/tex/hide) — vypíše výsledky výpočtů buďto v prostředí **Soutput**, nebo (**results=tex**) výsledky vloží jako čistý text bez jakéhokoliv prostředí, přesně tak, jak je vypisuje program R (vhodné např. pro předformátované tabulky viz str. 73), anebo nevypisuje žádný výsledek (**results=hide**).

```
<<results=hide>>= > x <- c(1,2,3,4)
x <- c(1,2,3,4) > mean(x)
mean(x)
@
```

label (text) — jeden z parametrů může být návěští pro blok. Návěští slouží k odkazům na obsah bloku na jiném místě dokumentu, nebo jako název pro obrázky v bloku vytvořené a pod. Odkaz na blok se vytváří posloupností znaků **<<label>>**.

Je-li **label** uveden jako první z parametrů, můžeme vynechat klíčové slovo **label** a psát jen samotné návěští.

```
<<a, echo=FALSE, results=hide>>=
x <- c(1,2,3,4)
mean(x)
@
<<>>= > 1+1
1+1 [1] 2
<<a>>
@ > x <- c(1,2,3,4)
> mean(x)
[1] 2.5
```

`keep.source` (FALSE/TRUE) — v základním nastavení Sweave upravuje kód pro R tak, že vynechá komentáře a změní řádkování a odsazení. V případě `keep.source=TRUE` přepisuje instrukce pro R přesně tak, jak jsou formátovány ve zdrojovém textu.

```
<<keep.source=FALSE>>          > x <- c(1, 2, 3, 4)
x <- c(1, 2,                    > mean(x)
      3,4)                      [1] 2.5
  mean(x) # výpočet
@
<<keep.source=TRUE>>          > x <- c(1, 2,
x <- c(1, 2,                    +   3,4)
      3,4)                      >   mean(x) # výpočet
  mean(x) # výpočet            [1] 2.5
@
```

`eval` (TRUE/FALSE) — povolí/zakáže vyhodnocení zdrojového kódu v R.

`split` (FALSE/TRUE) — uloží výstupy každý zvlášť do jednotlivých souborů, které jsou pak do hlavního souboru `*.tex` načítány příkazem `\input`. Název pomocných souborů je odvozen z názvu hlavního Rnw souboru, buďto přidáním čísel, nebo se za název připojí obsah proměnné `label` (viz obr. 5).

Pro globální změnu přednastavené hodnoty parametrů, použijeme v L^AT_EXovském zdrojovém souboru příkaz `\SweaveOpts`, v jehož argumentu jsou uvedeny příslušné parametry a jejich hodnoty (oddělené čárkou). Např.

```
\SweaveOpts{split=TRUE, keep.source=TRUE}
```

Tyto hodnoty parametrů se použijí ve všech následujících blocích kódu pro R.

Výsledky mezi textem v odstavcovém módu

Výsledky po vyhodnocení bloku jsou vkládány do textu mezi odstavce v prostředí Schunk. Pro vkládání jednotlivých čísel přímo do textu v odstavci slouží příkaz `\Sexpr{výraz v R}`. Argumentem tohoto příkazu je jediný výraz jazyka R, obvykle to bývá jedna proměnná, která je zavedena na jiném místě.

```
<<results=hide>>          > x <- c(1,2,3,4)
x <- c(1,2,3,4)          > m <- mean(x)
m <- mean(x)
@
Aritmetický průměr
vektoru  $x$  je \Sexpr{m}.
```

Uvedeno v souboru `prvni.Rnw`

```
<<split=TRUE>>=  
x <- 1:6  
mean(x)  
@  
  
<<sd-chunk, split=TRUE>>=  
x <- 7:10  
sd(x)  
@
```

Byl vytvořen soubor `prvni-01.tex`

```
\begin{Schunk}  
\begin{Sinput}  
> x <- 1:6  
> mean(x)  
\end{Sinput}  
\begin{Soutput}  
[1] 3.5  
\end{Soutput}  
\end{Schunk}
```

Byl vytvořen soubor `prvni-sd-chunk.tex`

```
\begin{Schunk}  
\begin{Sinput}  
> x <- 7:10  
> sd(x)  
\end{Sinput}  
\begin{Soutput}  
[1] 1.290994  
\end{Soutput}  
\end{Schunk}
```

V souboru `prvni.tex` je vloženo

```
\input{prvni-01}  
  
\input{prvni-sd-chunk}
```

Obrázek 5: Použití parametru `split`

Je důležité si uvědomit, že ačkoliv v R funkce `Sweave` vyhodnotí argument příkazu `\Sexpr`, funkce `Stangle` jej ignoruje. Zatímco při zpracování funkcí `Sweave` je tedy možné použít konstrukci

```
\Sexpr{(a <- 2+2)}
```

kteřá v R do proměnné `a` vloží součet, vyhodnotí a vypíše číslo 4, přičemž proměnná `a` bude k dispozici pro další výpočty na jiném místě, stejná konstrukce zpracovaná za pomoci `Stangle` bude vypisovat chybovou hlášku o chybějící proměnné `a`.

Obrázky

Obrázky a ilustrace vytvářené programem R je do $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ovského souboru možné začlenit volbou parametru `fig` a `include`.

`fig` (FALSE/TRUE) — příznak pro vytváření grafického výstupu. Názvy souborů s obrázky jsou odvozeny z názvu hlavního souboru přidáním čísel, nebo obsahu proměnné `label`.

`width`, `height` (6, číslo) — rozměry obrázku v palcích.

`pdf`, `eps` (TRUE/FALSE) — příznak pro typ grafického výstupu. Uplatní se jen, pokud je nastaveno `fig=TRUE`.

Ukažme si příklad bloku, jehož výsledkem je obrázek. (Výsledný obrázek viz obr. 6.)

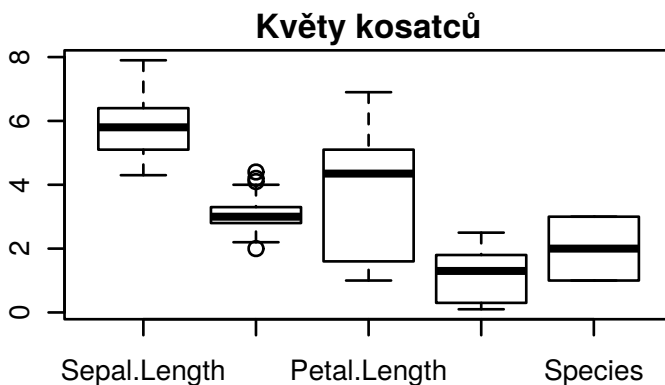
```
<<fig=TRUE, echo=FALSE, include=TRUE, width=3.5, height=2>>=
boxplot(iris, main="Květy kosatců")
@
```

Zařazení obrázků přináší dva hlavní problémy. Běžné utility programu R pro tvorbu obrázků ani do souborů PDF ani do PostScriptu nevkládají fonty, což ztěžuje přenositelnost výsledných souborů vzniklých kompilací $\text{T}_{\text{E}}\text{X}$ em. S tím okrajově souvisí druhý problém, nesoulad ve velikosti a typu písma v obrázku a v běžném textu. Obě tyto potíže lze překonat použitím vhodné grafické knihovny programu R. Grafickou knihovnu použitou funkcí `Sweave` nastavíme v parametru `grdevice`.

`grdevice` (text) — název funkce použité pro vytváření grafického výstupu ve `Sweave`. Je nezávislý na parametrech `pdf` a `eps`.

Jeden z dostupných grafických nástrojů v R je např. knihovna `Cairo`, jejíž použití s fonty `LatinModern` je v následující ukázce.

```
<<results=hide, echo=false>>=
require(Cairo)
```



Obrázek 6: Běžný grafický výstup z programu R

```
mainfont <- "lm roman 10"
CairoFonts(
  regular = paste(mainfont, "style=regular", sep=":"),
  bold = paste(mainfont, "style=bold", sep=":"),
  italic = paste(mainfont, "style=italic", sep=":"),
)

my.Swd.pdf <- function(name, width, height, ...) {
  CairoPDF(file = name,
    pointsize=10,
    width = width, height = height
  )
}

@
```

Následně je nutné nastavit parametr `grdevice`.

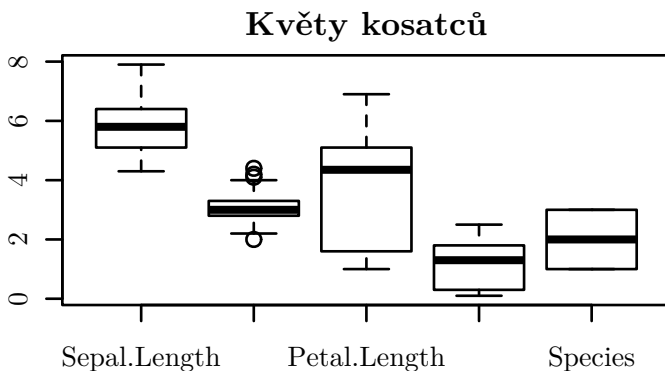
```
\SweaveOpts{grdevice=my.Swd.pdf}
```

V popiscích obrázku z předchozího příkladu jsou nyní nastaveny fonty LatinModern v příslušné velikosti (viz obr. 7).

Parametr `grdevice` můžeme použít i lokálně jako jeden z parametrů v jednom bloku.

V některých instalacích programu R je nejprve třeba LatinModern fonty *jednorázově* zavést do programu R.

```
> install.packages("extrafont")
> library(extrafont)
```



Obrázek 7: Obrázek s popisky ve fontu v LatinModern

```
> font_import(pattern = "lmodern*")
```

V každém následujícím použití programu R už budou LatinModern fonty k dispozici.

Obrázky jsou funkcí `Sweave` vygenerovány do pracovního adresáře a vkládány příkazem `\includegraphics` do \TeX ovského souboru bezprostředně na místo, kde je blok umístěn. Názvy obrázků jsou odvozeny z názvu hlavního `*.Rnw` souboru a pořadového čísla bloku, resp. z parametru `label`. Jiné umístění obrázku v textu umožní parametr `include`.

`include (TRUE/FALSE)` — vloží výstup bezprostředně na místo, kde je blok s kódem uveden, nebo v případě `FALSE` nevkládá `Sweave` do souboru `*.tex` nic a je možné vložit výstup manuálně.

Například v souboru `clanek.Rnw` uvedený blok

```
<<obr1, echo=FALSE, results=hide, include=FALSE>>=
par(mar=c(2,2,2,0))
boxplot(iris, main="Květy kosatců")
@
```

vytvoří obrázek s názvem `clanek-obr1.pdf`, ale nikam do \TeX ovského souboru nevkládá příkaz `\includegraphics`. To je nutné udělat ručně kdekoli v souboru `clanek.Rnw`.

```
\begin{figure}[ht]
  \includegraphics{clanek-obr1}
  \caption{Popis obrázku}
\end{figure}
```

Použití speciálního adresáře, do kterého se ukládají obrázky, umožní parametr `prefix.string`.

`prefix.string (text)` — obsah proměnné `prefix.string` se použije při tvorbě názvů souborů při nastavení `split=TRUE` nebo `fig=TRUE` namísto názvu hlavního souboru `Rnw`.

Nastavení `prefix.string=obrazky/fig` uloží všechny obrázky do adresáře `obrazky` a názvy obrázků budou složeny ze znaků `fig-` a návěští (resp. pořadového čísla) bloku. Adresář `obrazky` musí v souborovém systému existovat.⁴

Funkce `Sweave` nemá přehled o tom, jak grafická knihovna vytváří obrázky, takže není možné v jednom bloku vytvářet více obrázků najednou. Tj. konstrukce podobná následující není přípustná.

```
<<include=TRUE, fig=TRUE>>=
```

```
x <- rnorm(30)
```

```
boxplot(x)
```

```
hist(x)
```

```
@
```

Obrázky můžeme seskupit tak, že každý z obrázků vykreslíme ve zvláštním bloku s parametrem `include=FALSE` a vložíme je na správnou pozici pomocí `includegraphics` a možností `LATEXu`. Skupinu obrázků lze také vytvořit s využitím vlastností programu R pomocí parametru `mfrow`

```
<<include=TRUE, fig=TRUE, width=4, height=1.5, echo=FALSE>>=
```

```
x <- rnorm(30) # generujeme náhodná data
```

```
y <- rnorm(30)
```

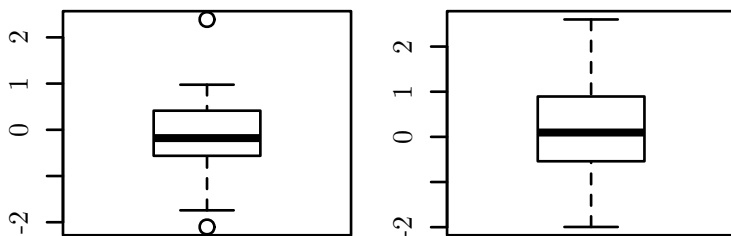
```
par(mfrow=c(1,2)) # obrázky zakreslíme do matice
```

```
                # s jedním řádkem a dvěma sloupci
```

```
boxplot(x)
```

```
boxplot(y)
```

```
@
```



nebo funkce `layout`.

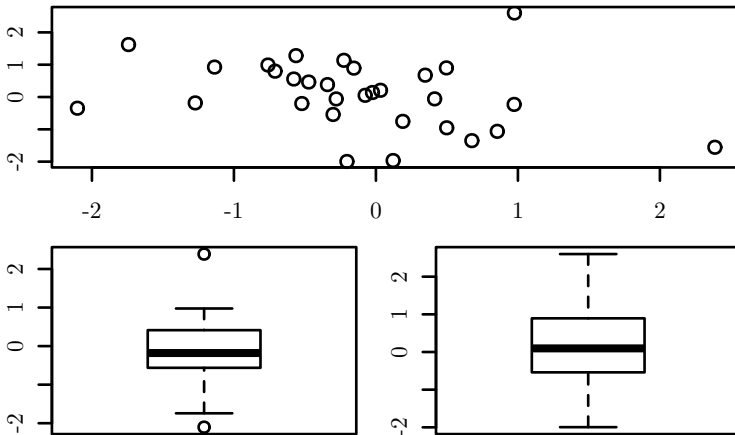
```
<<include=TRUE, fig=TRUE, width=4, height=2.5, echo=FALSE>>=
```

⁴Uživatelé `TEXu` nepřekvapí UNIXová notace pro oddělovače vnořených adresářů.

```

layout(matrix(c(1,1,2,3), byrow=TRUE, nrow=2))
# obrázky zakresluje do matice 2x2, kde první obrázek
# vyplňuje celý první řádek a druhý a třetí obrázek jsou
# umístěny vedle sebe na druhém řádku
plot(y ~ x)
boxplot(x)
boxplot(y)
@

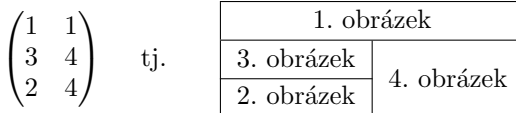
```



Jako další příklad použití funkce `layout` by mohlo sloužit např.

```
> layout(matrix(c(1,3,2,1,4,4), byrow=FALSE, nrow=3))
```

kterému odpovídá schéma obrázků o dvou sloupcích a tří řádcích vyplněné následovně



Jak pro parametr `mfrow`, tak pro funkci `layout` se redukuje velikost fontů v obrázku. Např. pro parametr `mfrow` programu R se v seskupeních 2×2 obrázků velikost fontu násobí koeficientem 0,83, pro tři a více řádků (sloupců) se použije koeficient 0,66.

Tabulky, matice

Pěkně zformátované tabulky a matice by měly být samozřejmostí v každém dokumentu vytvořeném \TeX em. Balíček `Sweave` žádným způsobem neřeší problém

vytváření tabulek a nechává tuto činnost jiným specializovaným utilitám. Přestože tento článek je věnován zejména balíku `Sweave`, je vhodné vytváření tabulek v programu R alespoň částečně nastínit.

V R je podporován výstup tabulek do $\text{T}_{\text{E}}\text{X}$ u několika balíky. Asi nejpoužívanější by mohl být balíček `xtable`. Funkce `xtable` objekty v R konvertuje na $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ovské tabulky v prostředí `tabular`. Lze takto např. v R vytvořit tabulku testu ANOVA (příkaz `aov`) a pomocí `xtable` ji konvertovat.

```
<<echo=FALSE, results=tex, include=TRUE>>=
library(xtable)
xtable(aov(Petal.Width ~ Species, data=iris))
@
```

Do $\text{T}_{\text{E}}\text{X}$ ovského souboru je vložen kód

```
% latex table generated in R 3.3.2 by xtable 1.8-2 package
% Tue Jan 3 17:17:03 2017
\begin{table}[ht]
\centering
\begin{tabular}{lrrrrr}
\hline
& Df & Sum Sq & Mean Sq & F value & Pr(>F) \\
\hline
Species & 2 & 80.41 & 40.21 & 960.01 & 0.0000 \\
Residuals & 147 & 6.16 & 0.04 & & \\
\hline
\end{tabular}
\end{table}
```

připravený pro kompilaci $\text{T}_{\text{E}}\text{X}$ em s výsledkem:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Species	2	80.41	40.21	960.01	0.0000
Residuals	147	6.16	0.04		

Funkcí `xtable` lze vytvářet i jiné objekty $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u než tabulky, např. matice. Využijeme parametr `tabular.environment` funkce `print.xtable`. Ve funkci `xtable` musíme vymazat výpis zarovnání sloupců (parametr `align`), zakázat výpis hlaviček u sloupců a řádků, zakázat mezirádkové linky a nedovolit, aby výsledný objekt byl vnořený do plovoucího prostředí.

`$$`

```
<<echo=FALSE, results=tex, include=TRUE>>=
matice <- matrix(1:9, nrow=3)
Xmatice <- xtable(matice, align=rep("", ncol(matice)+1))
print(Xmatice,
```

```

tabular.environment="pmatrix",
include.colnames=FALSE,
include.rownames=FALSE,
hline.after=NULL,
floating=FALSE)

```

@
\$\$

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

Problematika tabulek by zcela jistě zasloužila podrobnější pojednání.

Načítání dalších souborů

Načítání dalších *.Rnw souborů pomocí T_EXovského příkazu `\input` není možné, takto se dá načíst jen čistý T_EXovský kód bez bloků. Pro načtení `noweb` souborů, které budou předloženy programu R jako preprocesoru, je nutné použít sekvenci `\SweaveInput{soubor}`. Výsledky po zpracování programem R budou všechny vloženy do jediného souboru *.tex. Program R samozřejmě bere v úvahu změny proměnných, které jsou provedeny podle zadání v načítaném souboru.

Pokud příkazem `\SweaveInput` načítáme soubory, které nejsou jen čistý ASCII text, je třeba nastavit typ kódování při volání funkce `Sweave`, tj. použít `Sweave('soubor.Rnw', encoding='utf8')` nestačí jen použít v hlavním *.Rnw souboru L^AT_EXovskou sekvenci `\usepackage[utf8]{inputenc}`

Další nastavení

Funkce `Sweave` ukládá do souboru s příponou `tex` obsah a výsledky vyhodnocení bloku (s parametry `echo=TRUE`, nebo `results=verbatim`) obklopené prostředím `Schunk`, což je jen upravené prostředí `trivlist`.

Pro výpis kódu na vstupu a výstupu používá `Sweave` prostředí `Sinput` a `Soutput` založené na balíku `fancyvrb` a prostředí `Verbatim`. Jejich úprava je tedy pro uživatele jednoduchá. Např. nastavení výpisu výsledků v tomto dokumentu je

```

\DefineVerbatimEnvironment{Soutput}{Verbatim}{%
  fontshape=sl,
  xleftmargin=0pt,
  formatcom={\color{black}},
}

```

Pro přehlednější kód je dobré změnit chování R při výpisu rozdělených dlouhých vstupních řádků. V základním nastavení se v R pokračovací řádky na vstupu značí znakem +.

```
<<keep.source=TRUE>>=          > x <- c(1,2,
x <- c(1,2,                      +      3,4)
      3,4)
```

@

Lépe by bylo nastavit pokračovací prompt ve výstupech na (dvě) mezery. Kód bude čitelnější a v elektronické podobě lze kód bez pokračovacího promptu snadno kopírovat a spouštět v programu R.

```
<<echo=FALSE, results=hide>>=
options(continue = "  ")
```

@

a dostaneme

```
<<keep.source=TRUE>>=          > x <- c(1,2,
x <- c(1,2,                      3,4)
      3,4)
```

@

V českém jazykovém prostředí je také vhodné používat desetinou čárku namísto desetinné tečky, čehož docílíme nastavením

```
<<echo=FALSE, results=hide>>=
options(OutDec=",")
```

@

Pro uživatele „chytrých“ programů na prohlížení výstupů, propojených s editorem tak, aby na kliknutí myši editor nastavil zdrojový soubor na požadované místo (při kompilaci `pdflatex --synctex=1`), je určen parametr

```
\SweaveOpts{concordance=TRUE}
```

který uloží informace o tom, které řádky souboru `Rnw` odpovídají za kterou oblast souboru `tex`, aby propojení bylo funkční z prohlížeče až ke zdrojovému textu `Rnw`, a ne jen k mezistupni v souboru `tex`.

Závěr

Systém `Sweave` umožňuje psát dokumenty, články nebo zprávy, obsahující výsledky výpočtů, tabulky a obrázky, které dynamicky mění podobu v závislosti na vstupních datech. Pokud autor takových dokumentů dá komunitě k dispozici zdrojový kód dokumentu a vstupní data, je obrovskou výhodou možnost opakování výpočtů a ověření správnosti výsledků. Tento přístup je užitečný při týmové práci na vzniku dokumentu, kdy každý člen týmu má k dispozici na jednom místě celý

postup od dat až k finální podobě článku. Po publikaci by pak zveřejnění tohoto dokumentu a dat mohlo zprůhlednit vznik celé vědecké práce a usnadnit revizi závěrů. V neposlední řadě má také autor archivovány všechny kroky a postupy na jediném místě a nemusí spoléhat na vlastní paměť, odkud pochází ta která hodnota, tabulka, či obrázek v textu.

Reference

- GANDRUD, CHRISTOPHER. *Reproducible Research with R and R Studio*. 1. vyd. Boca Raton, Florida, USA : Chapman and Hall/CRC, 2013. 294 s. ISBN 1-466-57284-1.
- LEISCH, FRIEDRICH. *Sweave User Manual* [on-line]. 2009. [cit. 2016-12-20]. Dostupné na: <https://www.statistik.lmu.de/~leisch/Sweave/Sweave-manual.pdf>.
- LEISCH, FRIEDRICH. Dynamic generation of statistical reports using literate data analysis. In W. HÄRDLE, B. RÖNZ, Eds. *Compstat 2002 – Proceedings in Computational Statistics*. Heidelberg : Physica Verlag, 2002, s. 575–580. (ISBN 3-7908-1517-9.). Dostupné z DOI: 10.1007/978-3-642-57489-4_89.
- RAMSEY, NORMAN. Literate programming simplified. *IEEE software*, 1994, 11(5), s. 97–105. (ISSN 0740-7459.). Dostupné z DOI: 10.1109/52.311070.
- RSTUDIO TEAM. *RStudio: Integrated Development Environment for R* [on-line]. 2015. [cit. 2016-12-20]. Dostupné na: <http://www.rstudio.com/>.
- XIE, YIHUI. *Dynamic documents with R and knitr*. 2. vyd. Boca Raton, Florida, USA : Chapman and Hall/CRC, 2013. 294 s. ISBN 1-498-71696-2.

Poděkování recenzentům Autor by rád poděkoval oběma recenzentům za cenné podněty a připomínky, které pomohly tento text vylepšit.

Summary: Well-Documented Statistical Calculations

The paper describes the usage of the literate programming paradigm in the environment of statistical computations. In particular, the package **Sweave** for writing documents with the assistance of the statistical program R is presented.

Keywords: literate programming, reproducible research, Sweave, language R.

Marek Pomp, EkF VŠB-TU Ostrava, marek.pomp@usb.cz