

Zpravodaj Československého sdružení uživatelů TeXu

Marek Pomp

Tabulky v dobře dokumentovaných statistických výpočtech

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 28 (2018), No. 1-4, 22–37

Persistent URL: <http://dml.cz/dmlcz/150104>

Terms of use:

© Československé sdružení uživatelů TeXu, 2018

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 metodu publikování dobře dokumentovaných statistických výpočtů s pomocí programu R. Speciálně se věnuje vytváření tabulek za pomocí balíčků `knitr` a `kableExtra`. Článek volně navazuje na [4].

Úvod

Při psaní textů obsahující statistické výpočty je vhodné použít metodu tzv. *reprodukovatelného výzkumu* [2], to znamená publikovat výsledky výzkumu včetně získaných dat a přesného popisu metod při zpracování dat. Vytvářet a uchovávat text publikace, tabulky a ilustrace včetně přesného popisu použitých statistických výpočtů můžeme kombinací \LaTeX ovského zdrojového textu a zdrojového kódu v některém statistickém software, například v programu R. Jedna z možností, jak začlenit zdrojový kód jazyka R do \TeX ovského zdrojového souboru, použití balíku `Sweave` [3], bylo popsáno i v tomto časopise [4]. Přibližně v roce 2012 se balíčkem `Sweave` inspiroval Yihui Xie a začal s vývojem `knitru`, viz [6]. Na rozdíl od `Sweave`, který řeší jen export z R do \LaTeX u, je `knitr` obecnější a zvládá spolupráci programu R¹ s mnoha dalšími formáty dokumentů. V dalším textu bude stručně popsáno použití balíku `knitr` zejména s ohledem na sazbu tabulek v \LaTeX u.

Sazbu tabulek z programu R v \LaTeX u řeší velmi sofistikovaný balíček `xtable`. S rozšířenými možnostmi `knitru` je ale vhodné používat nástroje, které se nespecializují jen na tabulky ve formátu \LaTeX , ale dokáží vytvořit tabulky použitelné také v jiných značkovacích jazycích. Tyto podmínky splňuje funkce `kable` z balíku `knitr` podpořená rozšířením z balíku `kableExtra`.

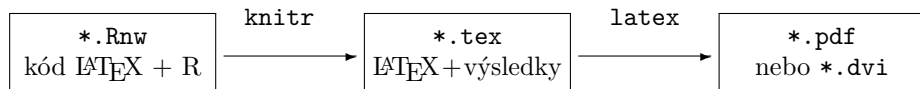
Poznámka pro čtenáře: Každá tabulka v textu je komentovaná a poblíž tabulky, obvykle na konci sekce, je uveden kód, jak příslušnou tabulku vytvořit. Naneštěstí není vždy možné, aby zobrazení tabulky, komentář a její kód byly na stejné straně.

Knitr – principy

Použití balíku `knitr` je v základě stejné jako u balíčku `Sweave`. Do \LaTeX ovského dokumentu se vkládají tzv. *chunks*, bloky kódu v jazyce R. Smíšený dokument s

¹Nejen R ale i jiných programovacích jazyků.

příponou `.Rnw` se kompiluje nejprve programem R. Během kompilace v programu R se vyhodnotí všechny bloky kódu jazyka R a výsledky se spolu s původním kódem \LaTeX ukládají do čistého \LaTeX ovského souboru. Tomuto postupu se říká *knitting* (v balíčce `Sweave` se tomuto procesu říkalo *weaving*). Z výsledného `*.tex` dokumentu lze vytvářet výstupy běžným \LaTeX ovským překladačem, viz obr. 1.



Obrázek 1: Schéma práce s `knitrem`.

Doplňkový proces nazvaný *purl* (pro `Sweave` se nazýval *tangle*) z původní směsi kódů pro program R a \LaTeX vybírá jen bloky s kódem pro R a ty ukládá do zvláštního souboru.

Bloky kódu programu R jsou ve zdrojovém dokumentu `*.Rnw` od \LaTeX ovského kódu odděleny hlavičkou ve tvaru

```
<<návěští bloku, volby a parametry bloku>>=
```

a za kódem v programu R následuje uzavírací znak bloku, zavináč `@`. Hlavička bloku i uzavírací zavináč musí být uvedeny na samostatném řádku, bezprostředně na jeho začátku (bez mezer a tabulátorů).

Soubor `*.Rnw` budeme v programu R kompilovat funkcí `knit` z balíku `knitr`. na příkazové řádce v linuxu² by příkazy pro kompilaci mohly vypadat následovně:³

```
$ Rscript -e "knitr::knit('soubor.Rnw'); pdflatex soubor.tex"
```

Pro větší komfort uživatelů programu R je v balíčce `knitr` také funkce `knit2pdf`, která použít postupně funkci `knit` a poté, aniž by musel uživatel opustit prostředí programu R, spustí rovnou překladač \TeX u zvolený v parametru `compiler`. Na příkazové řádce programu R bychom mohli spouštět např.

```
> knitr::knit2pdf("soubor.Rnw", compiler = "xelatex")
```

Funkce `knit2pdf` kompiluje překladačem \TeX u tolikrát, aby byly správně vyhodnoceny křížové odkazy, a pokud je to potřeba, zavolá v pravý čas také `Bi \TeX` a `MakeIndex`⁴.

Většina uživatelů používá k vytváření dokumentů nějaké komplexní vývojové prostředí, např. `Emacs`, `vim`, `WindEdt`, `\TeX nicCenter` a další.

²Příkazovou řádku operačního systému značíme promptem `$`, příkazovou řádku v programu R znakem `>`.

³Nebo v přehlednější formě

```
$ Rscript -e "require(knitr); knit('soubor.Rnw'); pdflatex soubor.tex"
```

⁴Funkce `knit2pdf` spouští \LaTeX prostřednictvím systémové funkce `latexmk`.

V komplexních editačních systémech je obvykle volba pro použití `knitru` přednastavená. Samozřejmostí je podpora `knitru` pro `LATEX`.

Protože kompilace celého dokumentu bývá zdlouhavá⁵, je vhodným zvykem ladit výsledné výpočty po částech, spouštět výpočet jen pro vybrané bloky kódu a celý dokument sestavit až pro správně probíhající výpočty. Tuto možnost nabízí např. editační prostředí RStudio (viz [5]), pluginy pro editory Emacs nebo vim a další nástroje.

Balík `knitr` není navržen jen pro spolupráci s `TEX`em, ale také pro psaní dokumentů v jazyce Markdown. Proto jsou některé jeho vlastnosti a funkce specializované na jeden z těchto jazyků a nemusí fungovat obecně. Autor textu, který nepočítá jen s využitím `knitru` v `LATEX`ovském dokumentu, by tedy měl mít na paměti, že je třeba zvážit, zda využít voleb a vlastností odkazujících na `LATEX`ovské speciality. Na tyto vlastnosti se pokusíme upozornit na příslušných místech.

Volby v hlavičce bloku

Volby v hlavičce bloku kódu určují chování programu R při vyhodnocování bloku a zobrazení obsahu bloku a jeho výsledků. Nejdůležitější volby pro bloky vytvářející tabulky jsou:

navěští bloku — je první⁶ volbou v hlavičce bloku a je to volba nepovinná.

Návěští, pokud je uvedeno, musí být pro každý blok jedinečné. Pro *navěští bloku* se doporučuje použít sekvenci alfanumerických znaků a pomlček, bez mezer. *Návěští bloku* slouží k odkazování na blok ve výpočtech, `knitr` umí vyhodnotit blok znovu na jiném místě dokumentu. *Návěští bloku* je také využito k pojmenování souborů s vygenerovanými obrázky. Dále je *navěští bloku* využito jako základ křížových odkazů na nadpisy tabulek. Křížový odkaz na tabulky bude mít v `LATEX`ovském souboru návěští označené `tab:navěští bloku` (obrázky mají automaticky vytvářené návěští `fig:navěští bloku`).

`echo` — určuje, zda se ve výsledném dokumentu objeví zdrojový kód zapsaný v bloku. Při čtení publikace asi čtenáře naše výpočty do detailu nezajímají a je vhodné nastavit tuto volbu na `FALSE` (a nechat zvědavé, ať detaily výpočtů studují ve zdrojovém kódu publikace).

`results` — popisuje, jakou formou budou zobrazeny výsledky. Protože výsledky výpočtů v tom formátu, v jakém je produkuje program R, nejsou pro

⁵Ti, kteří měli možnost překládat `TEX`ovské dokumenty na PC AT 286 rychlostí jedna strana za cca 15 s, mohou slovo „zdlouhavý“ relativizovat.

⁶`Knitr` umí v bloku volat i kód jiných jazyků, než programu R, takže úplně první volbou by mohl být název volaného jazyka např. `bash`, nebo `perl`.

nepoučeného čtenáře moc čitelné, formátujeme výsledky obvykle do vhodně popsaných tabulek. Pro tabulky exportované do formátu \LaTeX používáme volbu `results='asis'` (je možné použít i dvojité uvozovky "`asis`").

Celý blok tedy tedy může vypadat takto:

```
<<prvni_chunk, echo = FALSE, results = 'asis'>>=
... kód programu R
@
```

V dalším textu už bude vždy ukázán jen kód pro program R bez hlaviček bloku.

Volby pro hlavičku bloku lze nastavit globálně pro celý dokument. V některém z úvodních bloků může být uvedeno např.

```
knitr::opts_chunk$set(echo = FALSE, results = 'asis')
aby se v celém dokumentu neukazoval zdrojový kód, ale jen tabulky.
```

Tabulky obecně

Výstupy statistických výpočtů jsou nejčastěji prezentovány ve formě tabulek. Sazba tabulek je komplikovaná oblast typografie, ve které je velký prostor pro vedení sporů o to, co je správné. Přesto se asi většina sazečů shodne, že běžná tabulka z manuálů pro \LaTeX mívá

- příliš mnoho čar a linek,
- nepřehledné záhlaví a
- malou výšku řádků.

Vylepšení vzhledu tabulek řeší v \LaTeX u balíček `booktabs` [1]. V balíčku `booktabs` jsou zavedeny nové příkazy pro horizontální linky v tabulkách, je nastaven širší řádkový rejstřík tabulek a několik dalších drobností tak, aby tabulky vyhovovaly stylu University of Chicago Press.

Pro sazbu tabulek přesahujících jednu stranu je v \LaTeX u často používaný balíček `longtable`. Balíček `booktabs` je kompatibilní s balíkem `longtable`.

Jednoduché tabulky, funkce `kable`

Funkce pro tvorbu tabulek `kable` z balíku `knitr` [6] je velmi jednoduchá funkce a autor se podle vlastních slov nesnažil nahradit zavedený a sofistikovaný balíček `xtable` pro sazbu \LaTeX ovských tabulek v prostředí R. Funkce `kable` tak poskytuje jen základní možnosti sazby tabulek z objektů třídy `data.frame` nebo `matrix`⁷.

⁷Ve skutečnosti tato funkce dokáže vytvořit tabulky i z vektorů nebo objektů třídy `array` nebo `list`, ale tyto objekty je lépe předem upravit na obdélníková schémata typu `data.frame`.

Mějme jednoduchá data, několik řádků vybraných z datového souboru, který uvádí IQ dvojčat vychovávaných odděleně a jejich sociální status.

```
mydata <- carData::Burt[c(5,10,12,19,26),]
```

ze kterých vyrobíme tabulku funkcí `kable`.

```
kable(mydata, booktabs = TRUE,  
      col.names = c("První", "Druhý", "Třetí"),  
      caption = "Jednoduchá tabulka vytvořená funkcí \\texttt{kable}"  
    )
```

Tabulka 1: Jednoduchá tabulka vytvořená funkcí `kable`

	První	Druhý	Třetí
5	116	115	high
10	93	82	medium
12	88	100	medium
19	97	87	low
26	107	106	low

Prvním parametrem funkce `kable` je datový soubor. Parametr `booktabs` udává, zda má být tabulka formátována ve stylu *Chicago*, jen s vodorovnými linkami omezujícími tabulku a s vyznačeným záhlavím. Do záhlaví tabulky vstupuje parametr `col.names`. Pokud `col.names` není uveden, budou v záhlaví názvy sloupců z datového souboru. V tabulce budou uvedeny názvy řádků v závislosti na parametru `row.names`, přednastaveno je `row.names = TRUE`, takže se v tabulce objeví názvy řádků z datového souboru i když tato volba nebude uvedena.

Tabulka bude umístěna do plovoucího prostředí `table`, pokud obsahuje parametr `caption`. Návěští pro odkazy je automaticky přidáno podle parametru *návěští bloku* ve formě `\label{tab:návěští bloku}`. Na tabulku pak kdekoliv v textu odkazujeme pomocí `\ref{tab:návěští bloku}`.

Malá poznámka ke zdvojování zpětných lomítek. Většina poučených čtenářů ví, že zpětné lomítko bývá interpretováno jako speciální znak (escape), který mění význam pro ostatní znaky. Pokud chceme v regulárním výrazu vložit znak „zpětné lomítko“, zapisujeme jej jako escape sekvenci dvojicí zpětných lomítek. Příkazy \LaTeX u jsou proto v argumentech funkcí programu R uvozeny dvojicí zpětných lomítek. Ovšem, jestliže se regulární výraz vyhodnocuje vícekrát, je třeba znovu zdvojovat řetězec escape sekvencí, a proto např. v argumentu `big.mark` musí být malá \LaTeX ovská mezera zapsána pomocí čtyř zpětných lomítek a čárky, viz kód na str. 28.

Oddělení řádků mezerou, zarovnání sloupců

V tabulkách `kable`, které mají více než šest řádků, jsou, pro přehlednost, automaticky po pěti řádcích přidány vertikální mezery příkazem `\addlinespace`. Toto chování se dá změnit nastavením parametru `linesep`. Následující nastavení vkládá mezery už po třech řádcích.

Funkce `kable` má přednastaveno, že sloupce s čísly budou zarovnány vpravo a ostatní sloupce vlevo. Defaultní nastavení měníme v parametru `align` pomocí příznaků `c`, `r`, `l`. Tyto parametry můžeme zadat jako položky vektoru pro každý sloupec zvlášť, nebo zadáme jediný společný parametr pro stejné zarovnání všech sloupců. Ve druhém případě doopravdy využíváme schopnosti programu R recyklovat krátké vektory na potřebnou délku, takže např. pokud bychom chtěli zarovnat sudé sloupce tabulky jinak než liché, zadáme pouze dvě volby. Návěští řádků je zarovnáno vždy doleva.

V parametru `align` je možné přidávat také vertikální linky a další formátování sloupců podle zvyku \LaTeX ovských tabulek. S ohledem na využití tabulek také v dokumentech v jazyce Markdown je systémovější způsob namísto speciálních \LaTeX ovských sekvencí (např. `|` pro vertikální linku) použít pro formátování sloupců tabulky balík `kableExtra` a jeho funkce, např. funkci `column_spec` a její parametry `border_left`, `border_right`. Přesto je v následující tabulce poslední sloupec oddělen svislou linkou a formátován parametrem `p` na šířku 2 cm.

	IQbio	IQfoster	class
5	116	115	high
10	93	82	medium
12	88	100	medium
19	97	87	low
26	107	106	low

```
kable(mydata, booktabs = TRUE,  
      linesep = c("", "", "\\addlinespace"),  
      align = c("c", "l", "|p{2cm}")  
    )
```

Čísla v tabulkách

Číselné výstupy programu R jsou nastaveny tak, aby maximální počet číslic ve výpisech byl sedm, bez ohledu na desetinnou čárku. Změnit globální nastavení počtu číslic na výstupech je možné funkcí programu R `options` v parametru `digits`, např. `options(digits = 6)`

Tabulky vytvářené funkcí `kable` respektují nastavení počtu výstupních číslic, ale navíc je možné určit počet desetinných míst, pro zaokrouhlení výstupů v tabulce. Volba počtu *desetinných* míst pro zaokrouhlení se jmenuje `digits` (stejně jako globální volba počtu *všech* číslic na výstupu, což může být trochu matoucí).

Pokud bychom chtěli každý sloupec tabulky zaokrouhlit jiným způsobem, je možné ve volbě `digits` použít vektor. V následující tabulce je první sloupec zaokrouhlený na dvě a druhý na jedno desetinné místo.

Jako oddělovač desetinných míst slouží podle české typografické tradice desetinná čárka. Nastavit desetinnou čárku pro všechny výstupy lze globálně volbou `options(OutDec = ",")` Lokální nastavení v tabulce můžeme měnit v parametru `format.args` jako volbu `decimal.mark`.

	IQbio	IQfoster
5	36 442,47	36 128,3
10	29 216,81	25 761,1
12	27 646,02	31 415,9
19	30 473,45	27 331,9
26	33 615,04	33 300,9

Pro čtenáře je nepříjemné, pokud by měl číst dlouhá čísla, která postrádají jakékoliv formátování. Dobrým zvykem je oddělit každé tři číslice v číslovkách mezerou (nebo jiným znakem). Oddělovač tisíců se nastaví ve volbě `format.args`, parametrem `big.mark`. Protože argument funkce `big.mark` prochází dvojí expanzí, musíme zpětná lomítka zmnožit na čtyři. Funkce `kable` je nastavena tak, aby speciální znaky v buňkách expandovala do formátu stravitelného \LaTeX em. Např. zpětné lomítko transformuje na sekvenci `\textbackslash`, znak procenta na `\%` apod. Pokud ale vkládáme speciální \LaTeX ovské znaky přímo do buněk, musíme toto chování potlačit volbou `escape = FALSE`.

```
kable(mydata[c(1, 2)] * 100 * pi,
      booktabs = TRUE,
      digits = c(2, 1),
      escape = FALSE,
      format.args = list(decimal.mark = ",", big.mark = "\\\\",)
    )
```

Rozšíření funkce `kable`, balík `kableExtra`

Používání tabulek vytvářených funkcí `kable` se ujalo, (snad) hlavně proto, že `knitr` podporuje vytváření dokumentů v odlehčeném značkovacím jazyce `R Markdown`. Proto brzy vzniklo rozšíření možností pro formátování tabulek, balík `kableExtra`. Při použití balíku `kableExtra` by zdrojový text měl obsahovat řádek:

```
require(kableExtra)
```

Balík `kableExtra` zavádí několik nových funkcí, jejichž vstupem je tabulka `kable`, kterou na výstupu modifikují. Při práci s balíkem `kableExtra` se doporučuje využívat pro přesměrování výstupů softwarovou rouru, operátor `%>%`⁸. Roura

⁸Roura `%>%` není základní funkcí programu `R`, ale je balíkem `kableExtra` načtena z balíčku `magrittr`.

přesměruje výstup jedné funkce na vstup druhé. Např. v R (po načtení balíčku `kableExtra`) jsou tyto dva řádky

```
kable(mydata)
mydata %>% kable()
```

ekvivalentní způsob k zobrazení tabulky pomocí funkce `kable`.

Modifikace celých řádků a sloupců tabulky

Následující příklad ukazuje použití funkcí pro modifikaci řádků nebo sloupců `row_spec` resp. `column_spec`. Prvním z argumentů těchto funkcí je číslo řádku (sloupce), který máme modifikovat. Záhlaví je označeno jako řádek 0, ovšem názvy řádků, pokud se zobrazují, jsou počítány jako první sloupec.

Funkce `row_spec` a `column_spec` mohou modifikovat typ písma, přičemž názvy parametrů jsou samovysvětlující (**bold**, *italic*, `monospace`, underline, ~~strikeout~~⁹). Změna barvy písma a pozadí se provede nastavením parametrů `color` a `background`. Barvy lze zadávat ve formátu RGB nebo lze použít standardní názvy barev z L^AT_EXovského balíku `xcolor`. Balík `xcolor` je třeba mít načtený v hlavičce dokumentu, s příslušnou volbou (např. `x11names`).

Další parametry už mají funkce `row_spec` a `column_spec` rozdílné a zmíníme je pro každou funkci zvlášť.

Funkce `row_spec` může měnit velikost písma na vybraném řádku tabulky parametrem `font_size`. Tento parametr vyžaduje číslo, velikost fontu v bodech. Pro L^AT_EXovskou funkci `\fontsize`, která potom skutečně mění velikost fontu v řádku, je automaticky dopočítána výška řádku pro zvolenou velikost písma.

V parametru `align` je možné měnit zarovnání buněk v řádku písmenem `c`, `l`, `r`. Ve výsledném L^AT_EXovském souboru se tato volba projeví nastavením L^AT_EXovského makra `\multicolumn` pro všechny buňky na řádku (je tedy možné použít také volbu svislé čáry `|` pro vložení vertikální linky). Jestliže v této volbě uvedeme jediné písmeno, zarovnájí se stejně všechny buňky na řádku, vektor písmen může nastavit speciální zarovnání pro každou buňku. Volba `align` zruší speciální volby ve sloupcích. Na řádku 3 ve sloupci s

	IQ		
	<i>IQbio</i>	<i>IQfoster</i>	<i>class</i>
<u>5</u>	<u>116</u>	<u>115</u>	<u>high</u>
10	93	82	medium
12	88	100	medium
19	97	87	low
<u>26</u>	107	106	low

⁹Vyžaduje L^AT_EXovský balík `ulem`.

označením řádků tabulky podle nastavení sloupců by měla dvanáctka být vysázena kurzívou. Podobně je zrušena volba svislé linky za posledním sloupcem.

Parametr `angle` nastaví pootočení nápisu na řádku. Úhel otočení se uvádí ve stupních. Pootočení lze nastavit pro každou buňku řádku zvlášť tím, že uvedeme v parametru `angle` vektor.

Pokud je parametr `hline_after` nastaven na `TRUE`, bude za označeným řádkem vložena horizontální linka.

Pro sloupce jsou další volbou nastavenou ve funkci `column_spec` parametry `border_left` a `border_right` pro zobrazení vertikálních linek z jedné nebo druhé strany sloupce (to už umíme zařazením znaku `|` do volby `align` funkce `kable`, ale volby `border_*` jsou obecnější, použitelné nejen pro \LaTeX také pro jazyk Markdown).

Poslední funkcí použitou v tomto příkladu je `add_header_above`, která přidá další řádek nad běžné záhlaví tabulky. Také v záhlaví musíme mezi sloupce počítat i sloupec s názvy řádků, pokud je zobrazen volbou `row_names = TRUE`. Nové záhlaví bude odděleno od původního vodorovnou linkou. V \LaTeX u je tato linka vytvořená makrem `\cline`, které má v argumentu pomlčku, v české lokalizaci \LaTeX u aktivní znak. Před tabulkou musíme nastavit `\shorthandoff{-}` a za tabulkou zase vrátit znaku pomlčka aktivní příznak příkazem `\shorthandon{-}`.

```
kable(mydata, booktabs = TRUE, row.names = TRUE) %>%  
row_spec(0, angle = -45) %>%  
row_spec(1, underline = TRUE, strikeout = TRUE) %>%  
row_spec(2, bold = TRUE,  
  color = "white", background = "#959595") %>%  
row_spec(3, align = c("c", "l", "l", "r")) %>%  
row_spec(4, font_size = 8, hline_after = TRUE) %>%  
column_spec(1, italic = TRUE) %>%  
column_spec(4, border_right = TRUE) %>%  
add_header_above(c("", "IQ" = 2), bold = TRUE)
```

Formátování jednotlivých buněk

Speciální formátování některých vybraných buněk v tabulce se nastavuje funkcí `cell_spec`. Funkce `cell_spec` má podobné parametry jako `row_spec`, ale na rozdíl od funkcí `row_spec` a `column_spec` je třeba tuto funkci použít ještě před formátováním tabulky pomocí funkce `kable`. To znamená, že musíme modifikovat data, která poté budou zpracována funkcí `kable`. Protože vlastní data by měla zůstat po celou práci nedotčená, je vhodné pro zpracování do výstupů použít jen kopii dat.

V tabulce jsou buňky sloupce „IQfoster“ vysázeny menším osmibodovým písmem, pokud je hodnota v tomto sloupci menší než ve sloupci „IQbio“, jinak jsou vysázeny větším, dvanáctibodovým písmem. V této tabulce opět vkládáme speciální L^AT_EXovské sekvence, volbu fontu, přímo do buněk tabulky, takže musíme nastavit volbu `escape = FALSE`.

	IQbio	IQfoster	class
5	116	115	high
10	93	82	medium
12	88	100	medium
19	97	87	low
26	107	106	low

Funkce `cell_spec` mění datový soubor, proto musíme dodržovat pravidla programu R pro práci s datovými typy. Výsledek, který vrací funkce `cell_spec` je vždy typu „character“. Program R konvertuje na typ „character“ automaticky typ „numeric“, ale třeba typ „factor“ takové automatické konverzi nepodléhá. Protože třetí sloupec našeho datového souboru je zrovna typu „factor“, je nutné jej nejprve konvertovat na „character“, aby výstup funkce `cell_spec` byl kompatibilní s celým sloupcem. Použití funkce `cell_spec` na buňky prvního a druhého sloupce díky automatické konverzi nepůsobí problémy.

```
# nechceme měnit původní data, ale jen jejich kopii
cdata <- mydata
cdata$IQfoster <- cell_spec(cdata$IQfoster,
  font_size = ifelse(cdata$IQfoster < cdata$IQbio, 8, 12)
)
cdata[1,1] <- cell_spec(cdata[1,1], bold = TRUE)
# třetímu sloupci musíme změnit typ z faktoru na character
cdata[,3] <- as.character(cdata[,3])
cdata[1,3] <- cell_spec(cdata[1,3], bold = TRUE)
kable(cdata, booktabs = TRUE, escape = FALSE)
```

Poznámky v tabulce

Poznámky se do tabulky vkládají bez automatického propojení značky a textu poznámky, tj. vkládá se samostatně značka do tabulky a samostatně značka a text poznámky za tabulku. Značku pro poznámky do políčka nebo popisu tabulky vkládáme funkcemi `footnote_marker_*` (hvězdička nahrazuje volbu typu označení poznámky: `alphabet`, `number`, `symbol`, `none`). Značku do buňky v tabulce vložíme přímo do dat ještě před zpracováním tabulky funkcí `kable`. Samozřejmě, že připojení této značky k datům mění typ dat v buňce na typ „character“, a musíme stejně jako u funkce `cell_spec` věnovat pozornost tomu, aby s typem „character“ byl v souladu typ příslušného sloupce.

Parametrem funkcí `footnote_marker_*` je číslo, podle kterého se vybere

konkrétní značka pro daný typ markeru, např. pro marker typu `symbol` jsou prvními čtyřmi značkami znaky `*`, `†`, `‡`, `§`. V příkladu je nalezeno maximum a označeno poznámkou pod tabulkou. Další dvě poznámky jsou vloženy do buněk zadaných řádkovým a sloupcovým indexem.

Vlastní poznámky, tj. značku a text poznámky, vkládá funkce `footnote` za tabulku. Značky poznámek pod tabulkou se vyhodnocují v pořadí, v jakém jsou poznámky zapsány funkcí `footnote` a v závislosti na parametru `footnote_order`, který určuje pořadí pro zobrazení poznámek podle typu značky. V příkladu jsou nejprve zobrazeny poznámky značené písmeny a poté poznámka označená číslem.

Ve výsledném L^AT_EXovském souboru je poznámka za tabulkou vložena jako další řádek tabulky makrem `\multicolumn`. Protože šířka tabulky se řídí šířkou nejširšího řádku, což může být řádek s poznámkou, může dlouhá poznámka pod tabulkou ovlivnit šířku celé tabulky, jak tomu je v našem příkladu (viz tab. 2).

```
# nechceme měnit původní data, ale jen jejich kopii
cdata <- mydata
# nalezneme souřadnice pro maximum v numerických datech
maximum <- which(mydata[1:2] == max(mydata[1:2]), arr.ind=TRUE)
cdata[1:2][maximum] <- paste(mydata[1:2][maximum],
  footnote_marker_alphabet(2), sep="")
cdata[2,1] <- paste(cdata[2,1],
  footnote_marker_alphabet(3), sep="")
cdata[,3] <- as.character(cdata[,3])
cdata[1,3] <- paste(cdata[1,3],
  footnote_marker_number(1), sep="")
kable(cdata, booktabs = TRUE, row.names = TRUE, escape = FALSE,
  caption = paste("Poznámky v tabulce",
  footnote_marker_alphabet(1), sep="")) %>%
footnote(alphabet=c("Dlouhá poznámka způsobí roztažení tabulky",
  "Zde je maximum.", "Další poznámka v pořadí."),
  number = ("Poznámka označená číslem."),
  footnote_order = c("alphabet", "number")
)
```

Vylepšené chování poznámek v tabulkách zavádí L^AT_EXovský balík se jménem `threeparttable`. Tento balík vkládá poznámky pod tabulku do samostatné části, kterou zalamuje podle šířky tabulky. Balík `threeparttable` se aktivuje stejnojmennou volbou funkce `footnote`.

Funkce `footnote` umožňuje značit před poznámky také titulky.

	IQbio	IQfoster	class
5	116	115	high
10	93	82	medium

Číslované poznámky:

¹ Poznámka zalomená na šířku tabulky.

^a Další poznámka.

Tabulka 2: Poznámky v tabulce^a

	IQbio	IQfoster	class
5	116 ^b	115	high ¹
10	93 ^c	82	medium
12	88	100	medium
19	97	87	low
26	107	106	low

^a Dlouhá poznámka způsobí roztažení tabulky

^b Zde je maximum.

^c Další poznámka v pořadí.

¹ Poznámka označená číslem.

```
kable(mydata[1:2, ], booktabs = TRUE) %>%
footnote(threeparttable=TRUE, escape=FALSE,
  number = "Poznámka zalomená na šířku tabulky.",
  number_title = "Číslované poznámky:",
  alphabet = "Další poznámka."
)
```

Seskupení řádků, přídatné linky a zvýrazněné řádky

Řádky tabulky lze sdružovat funkcí `group_rows`. Její první parametr je označení skupiny řádků. Další dva parametry jsou číselné a ukazují na první a poslední řádek dané skupiny. V následujícím kódu jsou parametry funkce `group_rows` zadány buď napevno, nebo jsou vypočteny podle obsahu posledního sloupce tabulky.

```
kable(mydata, row.names=FALSE,
  booktabs = TRUE) %>%
group_rows("Medium class", 2, 3) %>%
group_rows("Low class",
  min(which(mydata$class == "low")),
  max(which(mydata$class == "low")))
```

IQbio	IQfoster	class
116	115	high
Medium class		
93	82	medium
88	100	medium
Low class		
97	87	low
107	106	low

Pokud je třeba v tabulce výrazně oddělit řádky, přidáme linky do funkce `row_spec` nastavením parametru `hline_after` resp. `extra_latex_after`. Už víme, že funkce `row_spec` má jako první parametr číslo řádku, na který se mají aplikovat dané speciality. Parametr `extra_latex_after` může obsahovat jakoukoliv sekvenci L^AT_EXovských příkazů, které budou vloženy za daný řádek. Pokud použijeme např. `\cline` pro L^AT_EX s českou volbou pro `babel`, musíme samozřejmě před tabulkou nastavit pomlčku z aktivního znaku na písmeno.

IQbio	IQfoster	class
116	115	high
93	82	medium
88	100	medium
97	87	low
107	106	low

```
kable(mydata, booktabs = TRUE,
      row.names=FALSE) %>%
row_spec(3, hline_after = TRUE) %>%
row_spec(1, extra_latex_after = "\\cline{1-2}")
```

Volba `latex_options = "striped"` pro funkci `kable_styling` barevně zvýrazní řádky tabulky. Toto zvýraznění provede příkaz `\rowcolors` z balíku `xcolor`, ve spolupráci s příkazem `\showrowcolors`. Oba tyto příkazy vkládá knitr do T_EXovského souboru za hlavičku tabulky. Barva zvýraznění je zvolena v parametru `stripe_color` a volí se z deklarovaných barev v závislosti na balíku `xcolor`.

IQbio	IQfoster	class
116	115	high
93	82	medium
88	100	medium
97	87	low
107	106	low

```
kable(mydata, booktabs = TRUE,
      row.names = FALSE) %>%
kable_styling(latex_options = "striped",
              stripe_color="gray!20")
```

Plovoucí prostředí pro tabulky

Do plovoucího prostředí `table` jsou tabulky umístěny, pokud obsahují parametr `caption` nebo pokud má funkce `kable_styling` nastavenou volbu `latex_options`. Parametry pro umístění tabulky na stránce jsou brány z defaultního nastavení L^AT_EXu¹⁰. Změnit nastavení pro umístění jediné tabulky můžeme funkcí `kable_styling` volbou `latex_options`. Parametr `[h]` bude zapsán nastavením `hold_position` a parametr `[H]` bude nastaven volbou `HOLD_position`. Pro druhou možnost potřebujeme mít v L^AT_EXu načtený balíček `float`.

¹⁰Globálně se tyto volby se mění definicí v hlavičce dokumentu, např. `\def\fps@table{htbp}`.

Tabulky obtékané textem zvládá L^AT_EXovské prostředí `wratable` z balíku `wrapfig`. Do tohoto prostředí se tabulka zapisuje volbou `position` nastavenou buďto na `float_right`, nebo `float_left`. Tabulka bude zařazena do následujícího odstavce. Pro prostředí `wratable` vynechává pro tabulku místo podle rozměrů tabulky.

IQbio	IQfoster	class
116	115	high
93	82	medium
88	100	medium

Obvykle je ale vhodné nastavit počet vynechaných řádků pro tabulku ručně. Např. předefinování `\def\WF@wli{7}` vyhradí pro tabulku místo vysoké jako sedm řádků. Samozřejmě musíme předem změnit kategorii pro zavináč na kategorii písmene makrem `\makeatletter` a potom zase zpět na kategorii makrem `\makeatother`.

```
kable(mydata[1:3,], booktabs = TRUE, row.names=FALSE) %>%
kable_styling(position = "float_right")
```

Naneštěstí funkce `kable_styling` zapíše pro prostředí `wratable` parametr pro šířku tabulky `Opt` (sázej tabulku podle její šířky), což působí kolize s nadpisem před tabulkou, a funkce `kable` neumí zařadit popisek až za tabulku, takže s popisky obtékaných tabulek nastávají potíže. Podobné potíže způsobí, jestliže byla při vytváření tabulky dvakrát použita funkce `kable_styling` (z toho jednou s volbou `latex_options`).

Tabulky s volbou `longtable`

Tabulky, které se rozměrem nevejdou na jedinou stranu, je vhodné sázet za pomoci prostředí `longtable`. V následujícím příkladu je tabulka se zvoleným prostředím `longtable`, ve které je přehlednost zajištěna podbarvením sudých řádků. Protože řádky jsou podbarveny, není zapotřebí oddělovat skupiny řádků separátorem `\addlinespace` z parametru `linesep`. Parametr `repeat_header_text` obsahuje text, který v záhlaví na dalších stranách oznamuje pokračování tabulky. Parametr `repeat_header_method` je defaultně nastaven na `append`, což znamená, že se na dalších stranách opakuje celý popisek z `caption`, nastavení `replace` zobrazí na další straně jen text z `repeat_header_text`.

Tabulka 3: Dlouhá tabulka

IQbio	IQfoster	class
82	82	high
80	90	high
88	91	high

Tabulka 3: Pokračování tabulky

IQbio	IQfoster	class
108	115	high
116	115	high
117	129	high
132	131	high
71	78	medium
75	79	medium
93	82	medium

```
mydata <- carData::Burt[1:10,]
kable(mydata, longtable = TRUE, linesep = "", booktabs = TRUE,
      caption = "Dlouhá tabulka") %>%
kable_styling(latex_options = c("striped", "repeat_header"),
              repeat_header_method = "replace",
              repeat_header_text = "Pokračování tabulky",
              stripe_color="gray!20")
```

Nastavení L^AT_EXu

V předchozím textu bylo zmíněno několik balíčků, které jsou zapotřebí pro sazbu tabulek produkovaných v programu R funkcí `kable`. Pro shrnutí jsou zde uvedeny balíky, které by při překladu dokumentu mohly být zapotřebí.

```
\usepackage{booktabs}      % tabulky oddělené linkami
\usepackage[table]{xcolor} % podbarvené řádky
\usepackage{makecell}     % speciální formát jednotlivých buněk
\usepackage{threeparttable} % poznámky pod tabulkou
\usepackage{float}        % volba [H] pro plovoucí tabulky
\usepackage{wrapfig}      % obtékané tabulky
\usepackage[normalem]{ulem} % přeškrtnuté písmo
\usepackage{longtable}    % dlouhé tabulky
```

Summary: Tables in well-documented statistical calculations

The article describes the method of publishing well-documented statistical calculations using software R. It is specially about creating tables using `knitr` and `kableExtra`.

Reference

- [1] FEAR, S. Publication quality tables in L^AT_EX, 2016. <https://ctan.org/pkg/booktabs>.
- [2] GANDRUD, C. *Reproducible Research with R and R Studio*. Chapman and Hall/CRC, Boca Raton, 2013.
- [3] LEISCH, F. Sweave: Dynamic generation of statistical reports using literate data analysis. In *Compstat 2002 — Proceedings in Computational Statistics (2002)*, W. Härdle and B. Rönz, Eds., Physica Verlag, Heidelberg, pp. 575–580. <http://www.stat.uni-muenchen.de/~leisch/Sweave>.
- [4] POMP, M. Dobře dokumentované statistické výpočty. *Zpravodaj Českoslvenského sdružení uživatelů T_EXu*, 1–4 (2016), 62–78.
- [5] RSTUDIO TEAM. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2015. <http://www.rstudio.com/>.
- [6] XIE, Y. *Dynamic documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, 2015.
- [7] ZHU, H. Create awesome latex table with knitr::kable and kableextra, 2018. https://haozhu233.github.io/kableExtra/awesome_table_in_pdf.pdf.

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

Marek Pomp, EkF VŠB-TU Ostrava