Úvod do R

Michal Kvasnička

V této úvodní lekci představíme R a jeho vývojové prostředí RStudio. Dozvíte se zde také, jak pracovat s knihovnami a jak hledat a číst dokumentaci k balíkům, funkcím a datům.

Co je R

R je jazyk a prostředí pro (statistické) výpočty a grafiku, více na https://www.r-project.org/about.html.

R umožňuje jak interaktivní práci (zadané výrazy se okamžitě vyhodnotí a výsledky vypíší na obrazovku), tak psaní skriptů (= programů, které je možné spouštět opakovaně na stejných nebo různých datech).

R je volně šiřitelný program pod licencí GNU. Je k dispozici pro Linux, Windows, Mac a další operační systémy (např. i Android).

Proč právě R

Existuje mnoho důvodů, proč používat R (a ne jiný software):

- R je volně šiřitelný software dostupný pro všechny hlavní operační systémy
- R je nejvíce používaný software pro data science
- R má velkou komunitu (často velmi pokročilých) uživatelů, takže
  - je dostupná spousta dobrých materiálů (knih, návodů na webu, video tutoriálů, kurzů na Coursera atd.)
  - vždycky vám někdo poradí
  - R obsahuje velké množství metod pro řešení téměř všech problémů v různých oblastech statistiky a data science
- veškeré potřebné nástroje jsou přítomné v jednom software (nemusíte používat jiný software na přípravu dat, jiný na jejich vizualizaci, jiný na ekonometrii, jiný na machine learning atd.)
- R obsahuje velmi mocný, ale relativně jednoduchý programovací jazyk, takže
  - můžete si napsat skript a svůj výpočet kdykoli zopakovat jeho spuštěním
  - svoji práci můžete zautomatizovat tak, že si na často opakované úkoly vytvoříte vlastní funkci nebo skript
  - pokud vám nějaká metoda v R chybí, můžete ji sami vytvořit a případně i sdílet s ostatními
  - svůj projekt vyřešit tak, že jej po vás kdokoli může zopakovat (podle zásad "reproducible research")
  - můžete smíchat výpočet a text a vytvořit "živé dokumenty"
- R je velmi rozšířené v akademické i komerční sféře

Rozšířenost R

Rozšířenost R pro datovou analýzu ukazují obrázky.

Více detailů např. zde: http://r4stats.com/articles/popularity/.

R se používá nejen na univerzitách; používá je i mnoho velkých komerčních firem, mimo jiné Microsoft, Facebook, Google, Twitter, Ford, Uber, John Deere, Firefox, The New York Times, The Human Rights Data Analysis Group a další, viz např. http://www.revolutionanalytics.com/companies-using-r a http://blog.revolutionanalytics.com/2014/05/companies-using-r-in-2014.html.

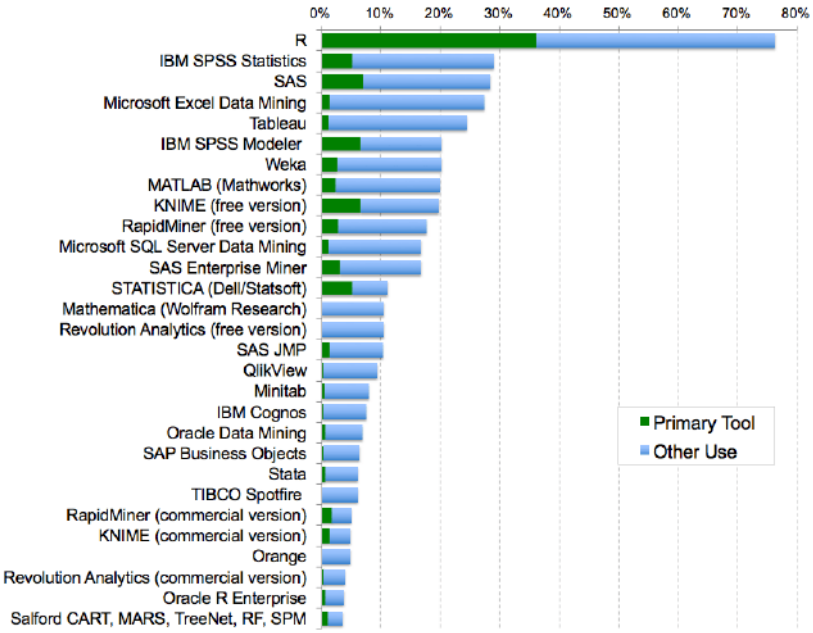


Figure 1: Analytické nástroje nejvíce používané respondnety Rexer Analytics Survey v roce 2015; každý respondent mohl zaškrtnout více nástrojů

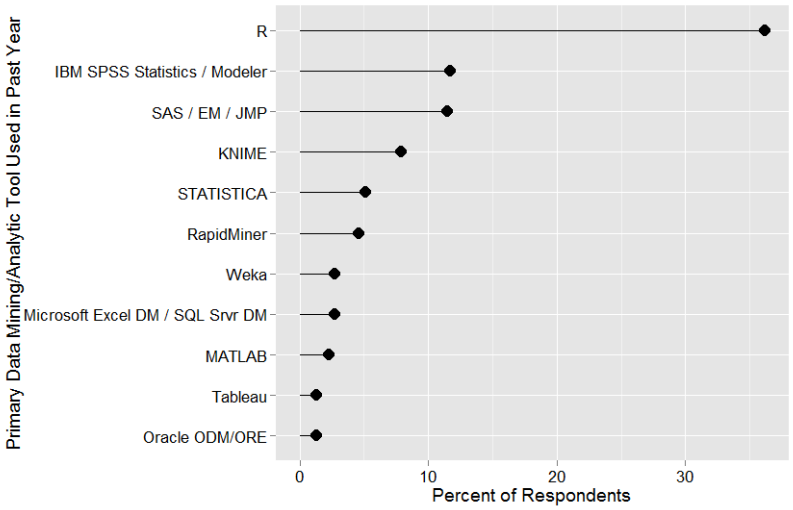


Figure 2: Primární analytický nástroj pro data science, stejný výzkum

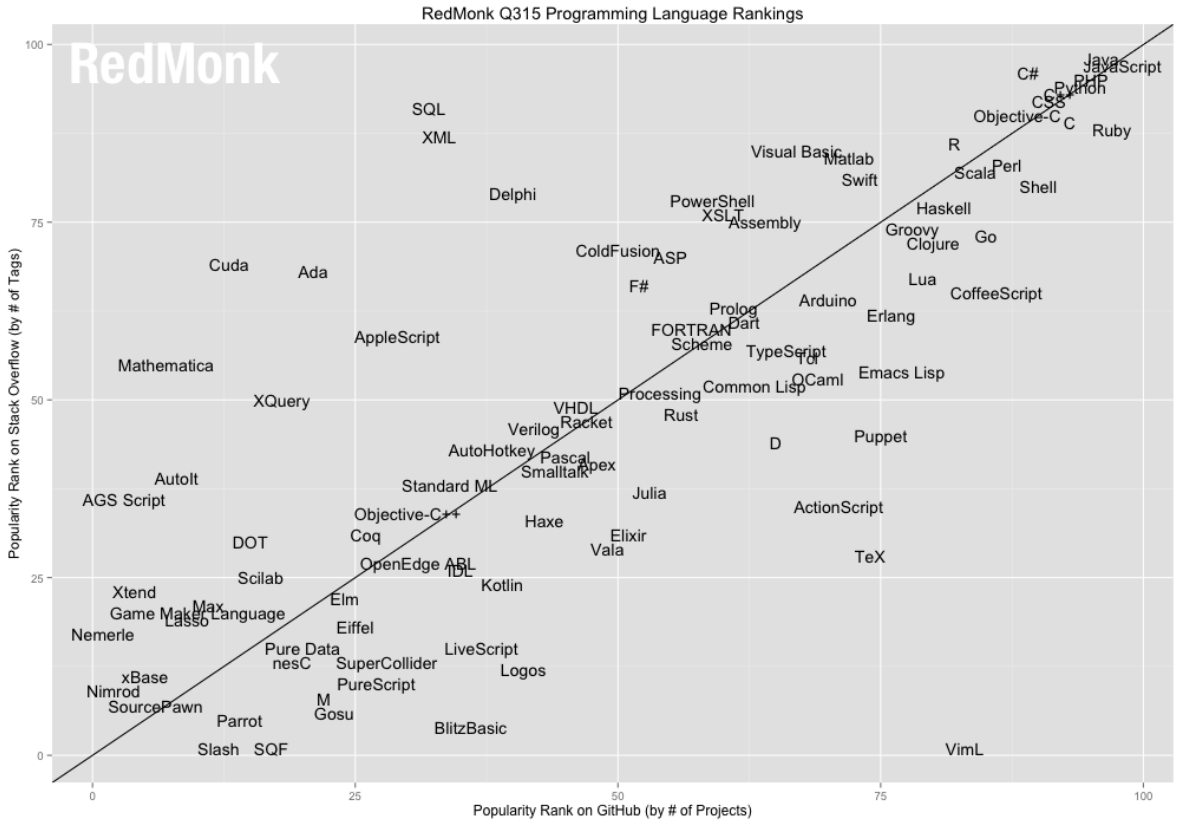


Figure 3: Popularita programovacích jazyků podle RedMonk 2015; na vodorovné ose pořadí podle počtu projektů na GitHub, na svislé ose pořadí podle počtu témat dotazů na Stack Overflow

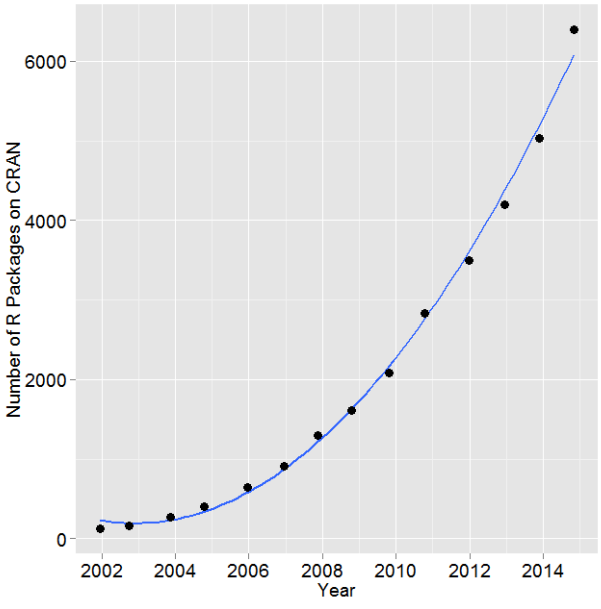


Figure 4: Počet balíku na CRANu ukazuje strmý růst objemu analytických nástrojů v R

Nevýhody R

R má samozřejmě i slabé stránky

- R je jazyk specializovaný na analýzu dat, nikoli obecný programovací jazyk – může se stát, že budete potřebovat i nějaký další speciální jazyk (Python, C++ apod.); na vlastní práci s daty je však R více než dostatečné
- R je poměrně pomalý jazyk; ve většině případů to však nevadí, protože funkce, které by v R běžely pomalu, jsou dávno implementované v C++ (případně Fortranu) a běží rychlostí těchto jazyků
- R dokáže pracovat pouze s daty, která má uložená v operační paměti počítače; je však nepravděpodobné, že v blízké budoucnosti budete mít tak velká data, že se do paměti nevejdou; R navíc dokáže spolupracovat s databázemi, a tak toto své omezení obejít
- R je poměrně starý jazyk, který prošel poměrně dlouhým vývojem; navíc vzniká decentralizovaně a na jeho vývoji se podílejí tisíce dobrovolníků; to kromě dobrých věcí zmíněných výše působí i některé problémy:
  - názvy funkcí a jejich syntaxe jsou někdy nekonzistenční
  - každou věc je možné udělat mnoha různými způsoby (a k řešení mnoha problémů existuje několik různých balíků funkcí)
  - chování některých funkcí je v některých speciálních situacích nečekané – kdysi to někomu přišlo jako dobrý nápad

Pořád je však podle mého mínění R nejlepší nástroj na analýzu dat na trhu.

Instalace R

Návod k instalaci a instalační soubory jsou ke stažení zdarma na https://cran.r-project.org/.

Windoze

Instalace ve Windows je jednoduchá. Mimo jiné ji ukazuje tento tutoriál: http://youtu.be/Ohnk9hcxf9M

Určité problémy mohou vzniknout, pokud máte v cestě ke svému domovskému adresáři mezery a písmena s háčky a čárkami. (Těm je však lepší se vyhnout vždy.) Pokud by standardní cesty kvůli mezerám, háčkům a čárkám nefungovaly, je možné nastavit jiné cesty, viz dále.

Linux

V mnoha distribucích Linuxu je R obsaženo přímo ve standardních repositářích, a to včetně knihoven. Doporučuji tyto balíky *neinstalovat*. Rozumnější je přidat si do repositářů **CRAN** a nainstalovat pouze jádro R a knihovny, které jsou tam v binární podobě. Ostatní knihovny si nainstalujete přímo v R. Tak budete mít vždy aktuální verzi systému. V Ubuntu vše funguje dobře.

V Linuxu používá R k maticovým výpočtům standardní numerickou knihovnu BLAS. Existuje několik verzí této knihovny, které se od sebe velmi výrazně liší výkonem – zejména tím, zda dokáží využít více jader procesoru. Doporučuji nainstalovat knihovnu OpenBLAS, která je v současnosti nejvýkonnější. Výrazně tak zrychlíte minimálně odhad ekonometrických modelů.

V Ubuntu stačí nainstalovat balíky `libopenblas-base` a `libopenblas-dev`. Nově nainstalovaný OpenBLAS by se měl automaticky použít. Která verze BLAS se použije, můžete ručně nastavit pomocí

```
sudo update-alternatives --config libblas.so.3 # vyberte OpenBLAS
sudo update-alternatives --config liblapack.so.3
```

Detaily viz http://blog.nguyenvq.com/blog/2014/11/10/optimized-r-and-python-standard-blas-vs-atlas-vs-openblas-vs-mk

RStudio

R funguje jako program na příkazovém řádku (ve Windows má k dispozici jednoduché grafické rozhraní). Pro vážnou práci s ním je však vhodné použít nějaké vývojové prostředí (IDE). Nejlepší IDE pro R na trhu je v současné době **RStudio**. Je volně šiřitelné pod licencí AGPL v3 pro Windows, Mac i Linux. Ke svému běhu potřebuje Javu (doporučuji oficiální Javu od Oracle, ne její svobodné ekvivalenty).

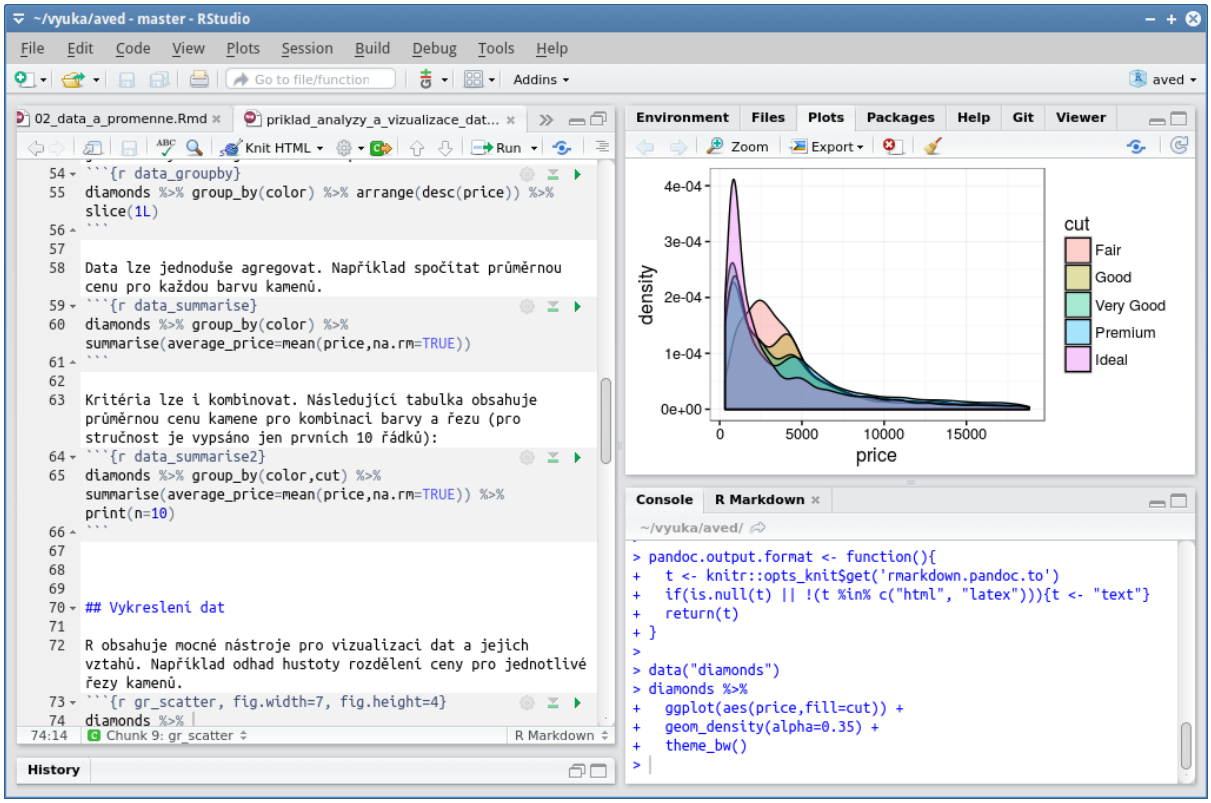


Figure 5: RStudio screenshot

Instalace RStudia

RStudio je zdarma ke stažení na https://www.rstudio.com/products/rstudio/download/. Pro Windows je dispozici instalační soubor, pro Linux standardní balíky `.deb` a `.rpm`.

Jednou za čas byste měli zkontrolovat, zda máte nejnovější verzi RStudia, a to v menu RStudia `Help→Check for Updates`.

Spuštění

R je interpretovaný jazyk, který je možné spustit (aspoň v Linuxu) přímo na příkazové řádce. My však budeme spouštět R v rámci RStudia. Při spuštění RStudia se v něm automaticky spustí i R. RStudio se spouští způsobem, který je v daném operačním systému obvyklý.

Hlavní panely RStudia

RStudio se skládá z obecného menu a čtyř panelů. Každý panel může obsahovat několik záložek. Můžete si zkonfigurovat, kde bude který panel, jak bude veliký a které záložky budou ve kterém panelu.

Hlavní záložky:

- **konzola** (Console) – kód, který zde napíšete, se v R okamžitě vyhodnotí a výsledky se vypíší do konzole
- **editor** (Source) – kód, který zde napíšete, můžete uložit a spouštět opakovaně (jako skript); v editoru můžete mít otevřeno libovolné množství souborů různých typů (R script, R markdown, textové dokumenty a mnohé další)
- **přehled prostředí R** (Environment) – zobrazuje všechny objekty (data, funkce apod.), které aktuálně žijí ve zvoleném prostředí v R (implicitně v globálním prostředí); umožňuje také importovat některé typy dat
- **soubory** (Files) – zobrazuje soubory a adresáře, které jsou v aktuálním projektu nebo adresáři a umožňuje s nimi dělat základní operace
- **grafy** (Plots) – zobrazuje grafy, které jste v R vykreslili
- **nápověda** (Help) – zobrazuje dokumentaci k funkcím, datům a knihovnám
- **balíky** (Packages) – zobrazuje seznam instalovaných knihoven; umožňuje také instalovat nové knihovny a staré knihovny mazat
- **historie** (History) – zobrazuje kód, který jste v minulosti spustili v konzoli; umožňuje jej také uložit a přesunout do konzoly a do editoru

Další typy záložek se objeví v případě, že budete dělat něco pokročilého.

Ikona *Workspace Panes* umožňuje jednotlivé panely a záložky dočasně zvětšit přes celou obrazovku.

V pravém horním rohu je přepínač projektů. **Projekty** umožňují elegantně oddělit vaše projekty – každý projekt má svůj adresář, vlastní proces R atd. Víc o projektech najdete zde: https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects.

Konfigurace RStudia

V menu RStudia `Tools→Global Options...` můžete nastavit, jak se má RStudio chovat.

Doporučuji zejména následující nastavení:

- záložka `General`: doporučuji vypnout `Restore .RData into workspace` a nastavit `Save workspace to .RData on exit` na `Never` – jinak se vám na začátku sezení nahrají do paměti výsledky výpočtů z minulosti; vypadá to jako dobrý nápad, ale je to zdroj chyb, které se špatně hledají
- záložka `Code→Editing`: zapněte `Insert spaces for tab` a `Tab width` nastavte aspoň na 4
- záložka `Code→Display`: zapněte vše (snad kromě `Highlight selected line`) a `Margin column` nastavte na hodnotu kolem 80
- záložka `Code→Saving`: zapněte vše
- záložka `Code→Completetion`: zapněte vše kromě `Use tab for multiline autocompletion`
- záložka `Code→Diagnostics`: zapněte vše
- záložka `Sweave`: nastavte `Weave Rnw files using knitr` a `Typeset LaTeX into PDF using` to `pdflatex`

Klávesové zkratky

Kromě menu můžete RStudio ovládat i pomocí klávesových zkratek. Seznam klávesových zkratek se v RStudiu zobrazí po stisku `Alt+Shift+K`. Úplný seznam klávesových zkratek najdete na https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts.

Ukončení

R lze ukončit funkcí `q()`. Pokud běží v RStudiu, ukončíte jej jednoduše buď v menu `File→Quit Session...` nebo křížkem okna.

R standardně při ukončení uloží všechny objekty v paměti (data, uživatelem definované funkce apod.) do souboru a při opětovném spuštění je opět načte (netýká se načtených knihoven – ty je třeba načíst pokaždé znovu). Někdy se to hodí, ale často je to zdrojem chyb, které se špatně hledají. Doporučuji tuto funkci v nastavení zakázat, viz výše.

Další zdroje k používání RStudia

Různé návody k používání RStudia najdete zde: https://support.rstudio.com/hc/en-us/sections/200107586-Using-RStudio.

Cheatsheet pro používání RStudia najdete zde: http://www.rstudio.com/wp-content/uploads/2016/01/rstudio-IDE-cheatsheet.pdf.

Knihovny

Všechny funkce, datové struktury, data i vše ostatní je v R organizováno do knihoven. Základní knihovny (`base`, `methods`, `datasets`, `utils` apod.) jsou přítomny v každé instalaci R a se načtou automaticky při jeho spuštění. Ostatní knihovny je potřeba nainstalovat a před použitím načíst do paměti.

Instalace a aktualizace knihoven

Seznam knihoven, které máte aktuálně nainstalované a načtené do paměti můžete v RStudiu zobrazit v záložce `Packages` (nahrané balíky mají zapnuté zaškrtávátko).

Většina balíků v R je k dispozici v centralizovaných repositářích. Hlavním repositářem je **CRAN** (https://cran.r-project.org/). Instalace balíků z CRANu je jednoduchá. V RStudiu stačí v záložce `Packages` kliknout na tlačítko `Install`. (Při první instalaci je třeba nastavit adresu zrcadla CRANu, ze kterého se budou balíky stahovat. Doporučuji použít zrcadlo `Global CDN RStudio`.) K ruční instalaci knihoven slouží funkce `install.packages()`.

Knihovny se vyvíjejí (většinou zlepšují) a obvykle je dobré mít instalovány poslední verze. K aktualizaci balíků slouží v RStudiu v záložce `Packages` klikátko `Update`. To však funguje jen pro knihovny nainstalované z CRANu.

Kromě CRANu existují i dva další velké centrální repositáře knihoven:

- **GitHub**, který obsahuje vývojové verze knihoven a nové knihovny, které se dosud nedostaly na CRAN, viz https://github.com/
- **Biocondutor**, který obsahuje převážně knihovny pro analýzu genomu, viz https://www.bioconductor.org/

Návod, jak instalovat knihovny z GitHubu, je zde: http://blog.numbersinlife.com/2012/12/installing-r-packages-from-github.html.

Poznámka: Knihovny se mohou instalovat bud do systémové části, nebo do uživatelského adresáře. Nevýhodou instalace do uživatelského adresáře je, že po přechodu na novou verzi R musíte své balíky nainstalovat znovu, protože se adresář s balíky zahrnuje v cestě i verzi R. (RStudio navíc zobrazuje zvlášť seznam balíků instalovaných v systémové a v uživatelské části.)

Existuje několik řešení:

1. V Linuxu můžete instalovat knihovny i do systémové části. K tomu stačí změnit přístupová práva příslušného adresáře (v Ubuntu při instalaci z CRANu `/usr/local/lib/R/site-library`) na `a+w`.
2. Ve Windows i v Linuxu můžete nastavit alternativní cestu k uživatelsky instalovaným balíkům (což se hodí i v případě, že standardní cesta nefunguje kvůli tomu, že obsahuje mezery, háčky a čárky). Postup je následující: 1) vytvoříte vhodný adresář (ve Windows např. `C:\my_R_packages`), kam můžete zapisovat; 2) do příslušného konfiguračního souboru přidáte na konec řádek

```
.libPaths(c("C:/my_R_packages"))
```

Pozor: místo zpětných lomítek používejte obyčejná lomítka (jinak musíte zpětná lomítka zdvojit, tj. psát `\\`). Můžete použít buď systémový konfigurační soubor `Rprofile.site` (při standardní instalaci je ve Windows umístěn v adresáři `C:\Program Files\R\R-X.X.X\etc`, kde `X.X.X` je číslo verze R, v Linuxu

v adresáři `/usr/lib/R/etc/Rprofile.site`), nebo osobní konfigurační soubor `.Rprofile`. K zápisu do systémového konfiguračního souboru potřebujete administrátorská práva.

Umístění nově instalovaných balíků pak můžete nastavit v RStudiu i ve funkci `install.packages()` (tam pomocí parametru `lib = "C:/my_R_packages"`).

Použití knihoven

Pokud máte nějakou knihovnu nainstalovanou, můžete ji začít používat. R však samo o sobě o objektech uložených v knihovnách neví. Před jejich použitím je třeba načíst balík do paměti (ve skutečnosti se jen načtou jména objektů v balíku do hledací cesty). K tomu slouží funkce `library()`:

```
library(dplyr)   # v závorce je jméno funkce
```

Při natažení nové knihovny se někdy stane, že nový balík překryje jméno funkce, která byla načtena v knihovně načtené dříve (R o tom vydá varování). I k překryté funkci je možné se dostat přímým voláním funkce, které má následující tvar:

```
dplyr::anti_join() # před dvojtečkami je jméno knihovny, za nimi jméno funkce
```

Stejným způsobem je možné volat i funkce z balíků, které nebyly načtené pomocí funkce `library()`.

Nápověda

Nikdo si nemůže pamatovat všechno – a pamatovat si detaily je absurdní. Proto má R velmi dobrý systém dokumentace.

Dokumentace k funkcím a datům

Nápovědu k přesné syntaxi funkcí můžete získat jedním ze tří způsobů:

```
?mean        # za otazníkem je jméno funkce
help("mean") # v uvozovkách je jméno funkce
```

nebo po napsání jména funkce zmáčknete v RStudiu klávesu `F1`.

Jedna stránka může dokumentovat několik různých funkcí, které mají něco společného. Stránky dokumentace mají v R standardní strukturu. Je potřeba, abyste se s ní seznámili. Stránka dokumentace má následující strukturu:

- název funkce / tématu a knihovny (např. `mean {base}` – `mean` je jméno funkce, `base` název knihovny)
- jméno stránky dokumentace
- popis, co funkce dělají (Description)
- syntaxe, jak se funkce používají (Usage) – tady se zejména píše, jaké má funkce parametry a jaké mají tyto parametry implicitní hodnoty (pokud nějaké mají)
- vysvětlení parametrů (Arguments) – zde je seznam všech parametrů funkce a vysvětlení, co parametr představuje a jakého má být daná proměnná typu
- hodnota funkce (Value) – zde se vysvětluje, jaké hodnoty funkce vrací, co znamenají a jakého jsou datového typu
- odkazy na literaturu (References)
- odkazy na jiné funkce, balíky nebo data (See Also) – odkazy na funkce, které nějak souvisejí s funkcemi, na které se právě díváte
- příklady použití funkcí (Examples)

Pokud otevřete dokumentaci v RStudiu, budou všechny odkazy klikací.

Dokumentace ke knihovnám

Dokumentaci ke knihovnám včetně seznamu funkcí, které jsou v ní obsažené, je možné získat dvěma způsoby:

```
help(package = "dplyr") # v uvozovkách je jméno knihovny
```

nebo v RStudiu tak, že v záložce Packages kliknete na jméno zvolené knihovny. V této záložce můžete i knihovny vyhledávat podle jména (ikona lupy v pravém horním rohu).

Demonstrační kódy

Mnoho funkcí a knihoven má k dispozici demonstrační kód. Tento kód můžete spustit takto:

```
demo("graphics") # parametr funkce je téma / jméno demonstrace
demo("bench-set", package = "dplyr") # pokud není knihovna načtená
```

Pokud chcete zjistit, jaké demonstrace obsahuje nějaká knihovna, zadejte

```
demo(package = "dplyr") # v uvozovkách je jméno knihovny
```

a RStudio otevře záložku se jmény demonstrací přítomných v daném balíku. Pak vyvoláte demonstraci obvyklým způsobem.

Viněty

K mnoha knihovnám existují **viněty**. Viněty jsou texty, které nedokumentují jednotlivé funkce, nýbrž ukazují, jak knihovnu použít jako celek nebo vysvětlují nějaký princip. Seznam vinět přítomných v daném balíku můžete zobrazit takto:

```
vignette(package = "dplyr") # v uvozovkách je jméno knihovny
```

Seznam vinět je také zobrazen v dokumentaci balíku, viz výše.

Jednotlivou vinětu můžete zobrazit buď pomocí

```
vignette("introduction", package = "dplyr") # první parametr je jméno viněty
```

nebo kliknutím na seznam vinět v dokumentaci ke knihovně v RStudiu.

Zdroje na webu

Jednou z výhod R je to, že má velkou bázi uživatelů. Proto je víc než pravděpodobné, že každý problém, který řešíte, už řešil někdo před vámi. Můžete tedy vyhledávat na Internetu, např. pomocí Googlu. Zkuste se podívat např. na výsledek hledání “nonparametric tests in r”.

Užitečné jsou také následující obecné weby:

- StackOverflow, http://stackoverflow.com/
- Cookbook for R, http://www.cookbook-r.com/
- Quick-R, http://www.statmethods.net/index.html
- R Bloggers, http://www.r-bloggers.com/
- R Tutor, http://www.r-tutor.com/
- obecné, někdy zbytečně technické originální manuály, https://cran.r-project.org/manuals.html

Knihy

Velmi dobré jsou následující knihy:

- Robert I. Kabacoff: *R in Action: Data analysis and graphics with R*, 2. vydání, Manning Publications, 2015 – velmi pěkný úvod do používání R
- John Verzani: *Using R for Introductory Statistics*, Chapman & Hall/CRC (The R Series), 2004 – pěkný úvod do použití R pro studium a použití statistiky
- Roger D. Peng: *R Programming for Data Science*, LeanPub, https://leanpub.com/rprogramming/ – docela dobrý úvod do R zhruba na úrovni našeho kurzu (jde legálně stáhnout zdarma)
- Roger D. Peng a Elizabeth Matsui: *The Art of Data Science: A Guide for Anyone Who Works with Data*, LeanPub, https://leanpub.com/artofdatascience/ – letmý úvod do Data Science (jde legálně stáhnout zdarma)
- Roger D. Peng: *Exploratory Data Analysis with R*, LeanPub, https://leanpub.com/exdata/ – jemný úvod do EDA (jde legálně stáhnout zdarma)
- Christian Kleiber a Achim Zeileis: *Applied Econometrics with R*, Springer (Use R), 2008 – přehled základních ekonometrických technik v R
- Hadley Wickham: *Advanced R*, CRC Press, 2014 – určeno pro čtenáře, kteří už R používají a chtějí se dozvědět víc do hloubky, jak R funguje

MOOCs

Coursera i Udacity mají několik kurzů analýzy a vizualizace dat v R.

Konzola

Konzola (Console) slouží k interaktivní práce s R. Všechny výrazy, které sem napíšete, se okamžitě vyhodnotí a výsledky se vypíší na obrazovku (do konzole).

Konzola tak slouží jako kalkulačka:

```
2 + 3 * 4         # výraz se okamžitě vyhodnotí
```

```
## [1] 14
```

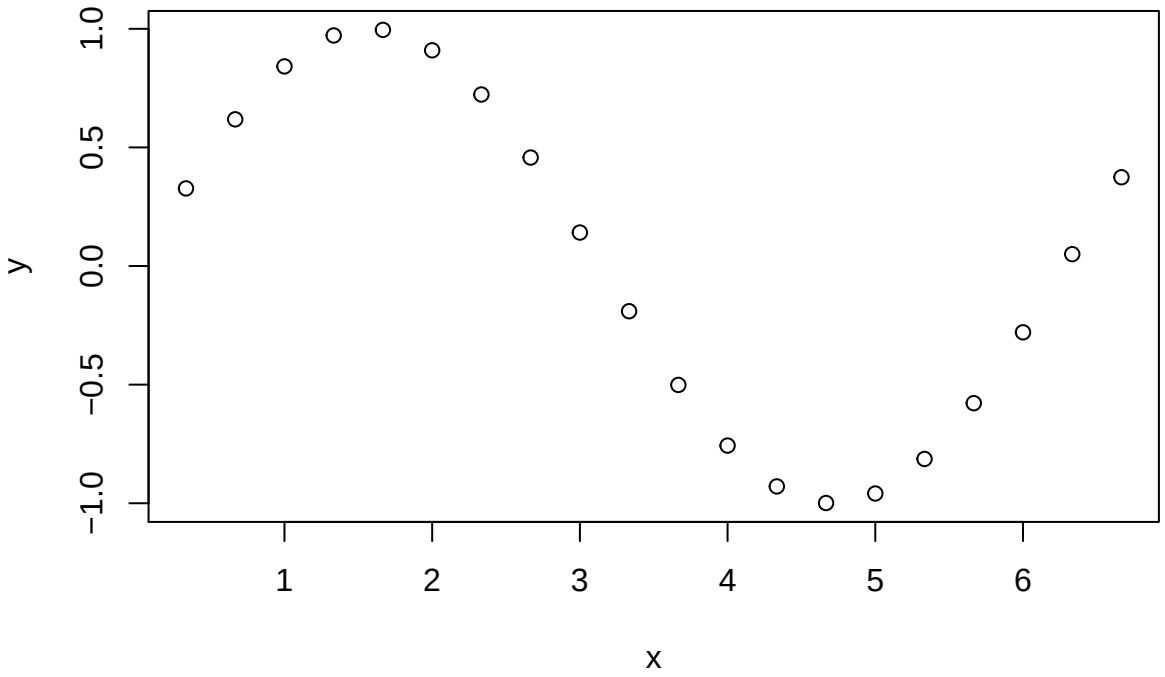
```
x <- (1:20) / 3   # přiřazení hodnot do proměnné x (nic nevypíše)
print(x)          # vypsání hodnot proměnné x
```

```
##  [1] 0.3333333 0.6666667 1.0000000 1.3333333 1.6666667 2.0000000 2.3333333
##  [8] 2.6666667 3.0000000 3.3333333 3.6666667 4.0000000 4.3333333 4.6666667
## [15] 5.0000000 5.3333333 5.6666667 6.0000000 6.3333333 6.6666667
```

```
x                 # totéž, co print(x)
```

```
##  [1] 0.3333333 0.6666667 1.0000000 1.3333333 1.6666667 2.0000000 2.3333333
##  [8] 2.6666667 3.0000000 3.3333333 3.6666667 4.0000000 4.3333333 4.6666667
## [15] 5.0000000 5.3333333 5.6666667 6.0000000 6.3333333 6.6666667
```

```r
y <- sin(x)       # vyhodnocení funkce a přiřazení do proměnné y  (nic nevypíše)
plot(x, y)        # vykreslení hodnot
```




```r
rm(x, y)          # vymazání proměnných x a y
# vše za symbolem křížku až do konce řádku je komentář -- R to ignoruje
```

Skripty

Skript je obyčejný textový soubor, do kterého napíšete R-kové výrazy jeden za druhý – každý nový výraz na nový řádek. Když pak skript spustíte, tyto řádky se provedou úplně stejně, jako byste je napsali přímo do konzoly. Vše na řádku za znakem křížku (`#`) se považuje za komentář; R tuto část řádku ignoruje.

Skripty je zvykem ukládat do souborů s koncovkou `.R`. Soubor se skriptem můžete vytvořit v jakémkoli textovém editoru, který k textu nepřidává žádné značky, tj. např. ne v MS Wordu. RStudio však poskytuje velmi dobrý editor, který umí barevně zvýraznit syntaxi, odhalit některé chyby, napovědět vám, jak se funkce jmenuje a jaké má parametry atd.

Kvůli ladění chyb i kvůli čitelnosti je dobré skripty pěkně formátovat. Doporučuji dodržovat např. styl Hadleyho Wickhama (http://adv-r.had.co.nz/Style.html) nebo (http://r-pkgs.had.co.nz/style.html) nebo styl Googlu (https://google.github.io/styleguide/Rguide.xml). RStudio vás dokáže upozornit na špatný styl, pokud si tuto volbu zapnete, a umí i částečně váš skript přeformátovat do pěknějšího (v menu `Code→Reformat code`); pomáhá také možnost automaticky odsadit řádky kódu (v menu `Code→Reindent lines`) a pěkně zarovnat komentáře (v menu `Code→Reflow comments`).

Jednou napsaný skript můžete spouštět znovu a znovu. Ke spuštění skriptu slouží funkce

```r
# jméno souboru do uvozovek
source("jmeno_skriptu_a_cesta_k_němu")
```

Funkce `source()` má mnoho dalších parametrů, viz dokumentace. Užitečný je zejména logický parametr `echo`, který ovlivňuje, zda se při spuštění skriptu vypisují do konzoly výrazy, které se právě vyhodnocují.

Skript, který máte otevřený v editoru v RStudiu spustíte snadno tím, že kliknete na tlačítko `Source` v pravém horním rohu editoru. I zde můžete nastavit, co se bude při zpracování skriptu vypisovat do konzole.

Někdy je užitečné spouštět i jednotlivé řádky skriptu otevřeného v editoru RStudia. Aktuální řádek nebo skupinu vybraných řádků spustíte klávesovou zkratkou `Ctrl+Enter`. Další možnosti spuštění skupin řádků a jejich klávesové zkratky najdete v menu `Code`.

Pár rad, jak se učit R

Nakonec pár poznámek k tomu, jak se učit pracovat s R.

1. Nesnažte se zapamatovat všechny detaily volání každé funkce – to snadno najdete v dokumentaci. Snažte se spíše pochopit princip, jak věci fungují.
2. Hrajte si a zkoušejte věci. Vymyslete si vlastní problém a zkuste jej vyřešit.
3. Kód z příkladů raději opište než kopírujte `Ctrl-C Ctrl-V`.
4. Vždy přemýšlejte, jak a proč něco funguje.
5. Když narazíte na neznámou funkci, přečtěte si pozorně dokumentaci a vyzkoušejte příklady na jejím konci.
6. Neděste se, že si nezapamatujete každý parametr každé funkce – to je normální a nevadí to, viz bod 1.
7. Když všechno selže, je tu Google.

Rady Hadleyho Wickhama

1. Čtěte zdrojový kód cizích funkcí. Tak se naučíte mnoho věcí.
2. Přistupujte ke kódu experimentálně. Pokud nevíte, jak něco funguje, vytvořte si hypotézu a vymyslete experiment, jak ji ověřit.

Dva slogany vysvětlující funogování R

Při práci s R vám pomůže, když budete mít "big picture", jak R funguje. To pěkně ilustruje následující citát:

> "To understand computations in R, two slogans are helpful:
>
> - Everything that exists is an object.
> - Everything that happens is a function call."
>
> — John Chambers

Domácí úkol

Soubor `income_data.RData` obsahuje vektor příjmů tisíce lidí. Vaším úkolem je spočítat průměrný příjem těchto lidí. K tomu slouží funkce `mean()`. Má to však dva háčky:

1. Některá pozorování v datech chybí (jsou nahrazena hodnotou `NA`). Vy chcete spočítat průměrný příjem těch lidí, pro která máte pozorování, tj. chcete vyloučit všechny hodnoty `NA`.
2. Většina lidí v datovém souboru má příjem mezi 10 a 30 tisíci Kč, data však obsahují i několik odlehlých pozorování lidí, kteří vydělávají 1 milion Kč měsíčně. Zahrnutí těchto "milionářů" by zkreslilo vypovídací schopnost průměru pro "obyčejné lidi", proto chcete vyloučit i je. Standardní řešení je tzv. rezistentní průměr, ze kterého se vynechá určitý počet nejnižších i nejvyšších pozorování. Vy chcete vynechat 5 % pozorování na obou stranách rozložení (myšleno 5 % pozorování *dohromady*).

Oba problémy umí funkce `mean()` vyřešit, pokud nastavíte správně příslušné parametry – detaily najdete v dokumentaci. (Pomoc: logické hodnoty jsou v R dvě: `TRUE` a `FALSE`.)

Výsledkem vaší práce bude upravený skript `hw_uvod_do_R.R`, který bude fungovat pro jakákoli data, která splňují výše zadané podmínky. Vaším úkolem je upravit jeden jediný řádek kódu. Ostatní řádky ani název souboru v žádném případě neměňte! Upravený soubor uložte do odevzdávárny "hw_uvod_do_R". Pamatujte, že se splnění úkolu bude testovat s jinými daty, než která máte zadaná jako vzor.

Poznámka: V případě příjmů je lepší rezistentní statistikou středu rozdělení medián. Tento úkol však vyžaduje výpočet rezistentního průměru.