

Úvod do jazyka R, zpracování a vizualizace dat

Jan Caha

Interpretace geodat - 1. blok
8. 3. 2018

1 Úvod

2 Základy jazyka R

3 Operace s daty

4 Vstup a výstup dat

5 Tvorba grafů

Něco o mně

- studium: na katedře Geoinformatiky, Univerzita Palackého v Olomouci (Bc., Mgr., Ph.D, RNDr.)
- praxe: VŠB-TUO, Mendelova Univerzita v Brně
- odborné zájmy: zpracování nejistoty v GIS, fuzzy logika a fuzzy arimetika, analýzy viditelnosti, data science (se zaměřením na prostorová data/analýzy), open source (obecně + GIS)
- jazyky:
 - dříve: Java, C#
 - aktuálně: R, Python
- moje práce:
 - Github
 - ResearchGate

Motivace

- jako geoinformatik věnujete 20-80% pracovního času zpracování dat
- každou analýzu budete 5x-10x předělávat, než se dopracujete ke konečnému výsledku
- rozšiřování obzorů, zkušeností, schopností, znalostí a karierních možností
(zajímavé čtení např. [zde](#), [zde](#), a [tady](#))

Jazyk R

- implementace jazyka **S** pod open source licencí
- určený primárně ke zpracování dat a jejich vizualizaci, s výrazným zaměřením na statistiku
- interpretovaný programovací jazyk přístupný v základu především skrze příkazový řádek
- základní funkcionality není příliš široká, velká část funkcionality přichází z tzv. balíčků
- centrální sklad balíčků - **CRAN** (Comprehensive R Archive Network)
- **CRAN** je také místem pro stažení samotného R a balíčků

Práce s R

- většinou nepracujeme přímo s příkazovým řádkem R
- využití IDE (Integrated Development Environment), které poskytuje nástroje pro signifikantní zvýšení komfortu práce s R
- řada IDE zaměřených na R - [Rattle](#), [RKWard](#), [R Commander](#)
- nicméně téměř standardem a nejpopulárnějším řešením je [RStudio](#)
- R lze používat několika způsoby
 - tzv: interaktivní mód, kdy pracujeme "na živo" s daty
 - programování analýz
 - programování balíčků
- nejčastější je kombinace interaktivního módu s programovaním analýz (to budeme předpokládat)

RStudio

- popis prostředí RStudio

The screenshot shows the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins. The main window has several panes:

- Code Editor:** Displays the file "predmesta-1.Rmd" with R code. The code discusses the history of R, its open source license, and various R environments like Rattle, R Commander, and R Studio.
- Environment Browser:** Shows the Global environment, which is currently empty.
- PLOTS, PACKAGES, HELP, VIEWER:** Standard RStudio navigation tabs.
- Help Viewer:** Displays information about the "gapminder" dataset, including its description, usage, and format.

The bottom pane shows the R console output, which includes:

- Information about RStudio Community and how to cite R or R packages.
- Instructions for running demos, getting help, and quitting R.
- Output from running R code to install the "gapminder" package from CRAN.
- A message indicating the package was successfully unpacked and MDS sums checked.

Historie R

- vývoj jazyka začal v roce 1993
- veřejně představen v roce 1996 (Ihaka and Gentleman, 1996)
- hlavní verze 1.0 (2000), 2.0 (2004), 3.0 (2013), aktuálně 3.4.3
- RStudio - první veřejné verze (2010/2011) - značný dopad na komunitu uživatelů R, profesionalizace vývoje některých (důležitých) balíčků
- cca or roku 2014 nesmírně dynamický vývoj do ruzných oblastí využití R
- aktuálně na CRAN dostupných přes 12 000 balíčků

Vztah R k prostorovým datům a analýzám

- primární účel - statistická analýza geodat
- první pokusy už v roce 2000 (Bivand, 2000)
- v roce 2003 už existovalo minimálně 21 balíčků zaměřených na prostorová data (Bivand, 2003)
- formální definice prostorových objektů v R až s balíčkem `sp` (Pebesma and Bivand, 2005), který se stal de facto standardem
- později balíček `raster` (Hijmans, 2017)
- aktuálně snaha nahradit `sp` za nový balíček `sf` (Pebesma, 2017), který vhodněji odpovídá aktuálním standardům práce s daty v R

Data Science

- datová věda
- definice: interdisciplinární obor propojující vědecké metody, procesy, algoritmy a systémy za účelem extrakce znalostí a pohledů na data, ať už strukturovaná či nikoliv (Dhar, 2013)
- většinou značnou část práce zahrnuje předzpracování, zpracování a úprava dat (data wrangling)
- další činnosti jsou obvykle vizualizace a prezentace dat
- právě data science je aktuálně hlavní hybnou silou ve vývoji R

Zdroje informací

- nápověda R + viněty
- knihy např. oficiální seznam na *webu*
- seznam volně dostupných (open source) knih týkajících se R - *bookdown*
- *R4DS, Advanced R, R Programming for Data Science, Spatial Microsimulation with R* atd.
- blogy - např. *R-bloggers*
- twitter - obzvláště např. *Mara Averick*, ale i větší množství dalších
- *Stack overflow* a jeho varianty

Základní příkazy pro další práci

- instalace a načtení balíčku

```
install.packages("nycflights13")
```

```
library(nycflights13)
```

```
install.packages(c("ggplot2", "dplyr", "gapminder"))
```

```
library(ggplot2)
```

```
library(dplyr)
```

Proměnné a hodnoty

- tvorba proměnných
- přiřazování hodnot
- vektory proměnných, seznamy proměnných

```
promenna <- 1
promenna_1 <- 3.1415926
promenna_2 <- "textova proměnná"
promenna_3 <- TRUE

vektor_promennych <- c(1, 4, 7, 2, 5, 6, 9, 15)
seznam_promennych <- list("a", 5, 8.45)
```

Operace s proměnnými

- veškeré matematické operace

```
x <- 2
```

```
y <- 4
```

```
vector_x <- c(5, 7, 2)
```

Operace s proměnnými

```
x + y
```

```
## [1] 6
```

```
x - y
```

```
## [1] -2
```

```
x * y
```

```
## [1] 8
```

```
x / y
```

```
## [1] 0.5
```

Operace s proměnnými

```
z <- x * y
```

- vektorizované verze oprací

```
vector_x + vector_x
```

```
## [1] 10 14  4
```

```
vector_z <- vector_x * vector_x  
vector_z
```

```
## [1] 25 49  4
```

Funce proměnných

- vektorizované verze
- proměnné bez jména proměnné i pojmenované proměnné

```
sin(x)
```

```
## [1] 0.9092974
```

```
log(y, base = 3)
```

```
## [1] 1.26186
```

```
cos(vector_x)
```

```
## [1] 0.2836622 0.7539023 -0.4161468
```

Komplexní datové struktury

- 2D struktury pro uložení dat
- matice
- datové rámce (data frames)
- v maticích musí být všechny prvky stejného typu, v datovém rámci se mohou jednotlivé sloupce lišit
- pro zpracování dat se obvykle používá data.frame, či jeho modernější verze tibble

Ukázka datového rámce

```
data(flights)
View(flights)
data(diamonds)
View(diamonds)
```

Faktory

- proměnné, které by teoreticky mohly být textové, ale chceme u nich mít omezený obor hodnot, nebo např. pořadí prvků
- viz dataset diamonds - proměnné cut, color, clarity
- více o faktorech později

Operace s daty

- ačkoliv lze primárně veškeré operace provádět pouze s použitím základního R, je mnohem uživatelsky přívětivější používat funkce balíku dplyr
- dplyr umožňuje i příhodně řetězit operace za sebe, vytváření tzv. pipelines, pomocí operátoru `%>%`

Získání části dat

- v rámci data frame (tibble) se lze odkazovat a provádět výběry mnoha způsoby
- lze se odkazovat na řádek/řádky, sloupec/sloupce, konkrétní buňku/buňky či na řádky splňující učitou podmínu

```
# řádek  
diamonds[1,]  
  
# sloupec  
diamonds$carat  
  
# buňka  
diamonds[1,2]  
  
# několik řádků a sloupců  
diamonds[c(1,5,7), c(1,2,3)]  
  
# výběr podmínkou  
diamonds[diamonds$color == "E", ]  
diamonds[diamonds$price > 18800, ]
```

Balík dplyr

- určeno pro práci s data data frame (Wickham *et al.*, 2017)
- “slovesa” pro práci s daty, značně zjednodušuje práci s daty
- velice efektivní kód na pozadí, díky kterému jsou i náročné operace velice rychlé

```
?dplyr
```

```
## starting httpd help server ... done
```

- ukážeme si jen nejvýznamější funkce a postupy, ale balík poskytuje obrovskou škálu nástrojů
- zápis funkcí, buď jednoduše nebo spojení do pipeline

```
filter(diamonds, price > 18800)
# s využitím pipeline
diamonds %>% filter(price > 18800)
```

Filtrování dat dle pořadí

- funkce `slice`
- záporné hodnoty vynechají dané prvky z výběru

```
diamonds %>%  
  slice(1)
```

```
diamonds %>%  
  slice(1:5)
```

```
diamonds %>%  
  slice(-5:-nrow(diamonds))
```

Filtrování dat dle podmínky

- funkce filter
- odělení čárkou znamená použití podmínky AND, nebo lze místo čárky použít symbol &
- podmínka OR se zapisuje pomocí symbolu |

```
diamonds %>%
  filter(price > 18800, cut == "Ideal")
```

identický zápis

```
diamonds %>%
  filter(price > 18800 & cut == "Ideal")
```

```
diamonds %>%
  filter(price > 18800 | cut == "Ideal")
```

Filtrování dat dle podmíny

- komplexní dotazy

```
diamonds %>%
```

```
  filter((price > 15000 | cut == "Ideal") & carat > 3)
```

```
## # A tibble: 23 x 10
##   carat cut      color clarity depth table price     x     y     z
##   <dbl> <ord>    <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 3.22 Ideal     I      I1     62.6  55.  12545  9.49  9.42  5.92
## 2 3.50 Ideal     H      I1     62.8  57.  12587  9.65  9.59  6.03
## 3 4.01 Premium   I      I1     61.0  61.  15223 10.1   10.1   6.17
## 4 4.01 Premium   J      I1     62.5  62.  15223 10.0   9.94  6.24
## 5 3.04 Very Good I     SI2     63.2  59.  15354  9.14  9.07  5.75
## 6 3.40 Fair      D      I1     66.8  52.  15964  9.42  9.34  6.27
## 7 4.00 Very Good I      I1     63.3  58.  15984 10.0   9.94  6.31
## 8 3.01 Ideal     J     SI2     61.7  58.  16037  9.25  9.20  5.69
```

Výběr x prvků podle hodnoty ve sloupci

- funkce `top_n`
- nemusí vracet n prvků, pokud je shoda na posledním místě, může být prvků více
- lze využít i pro výběr minima, bud' uvedením negativní hodnoty počtu prvků nebo s využitím funkce `desc` na výběrovém sloupci

```
diamonds %>%
  top_n(5, carat)
```

```
# výběr minima
diamonds %>%
  top_n(-10, price)
```

```
diamonds %>%
  top_n(10, desc(price))
```

Výběr pouze specifických sloupců z datasetu

- funkce `select`

```
data_pro_dalsi_analyzu <- diamonds %>%  
  select(color, cut, price) %>%  
  filter(price > 15000)
```

Vytvoření nového sloupce

- funkce *mutate*
- ve vzorci lze používat libovolné funkce R

```
diamonds_new <- diamonds %>%  
  mutate(price_per_carat = price / carat)
```

v názvech proměnných v data frame lze použít i mezery a další znaky,
ale pak je nutné je vždy ukládat v ''
diamonds_new <- diamonds %>%
 mutate('price per carat' = price / carat)

Seřazení data frame

- funkce `arrange`
- lze uvést více sloupců, řadí se podle pořadí, další mají roli v případě shody v předchozích

```
diamonds_new <- diamonds_new %>%  
  arrange(desc(price_per_carat))  
View(diamonds_new)
```

```
diamonds_new <- diamonds_new %>%  
  arrange(desc(price), cut)  
View(diamonds_new)
```

Vytvoření skupin v datech

- funkce `group_by`
- určujeme podle kterých sloupců májí být vytvořený skupiny
- samotná funkce na data "nemá vliv", alespoň z vizuálního hlediska, nicméně z funkčního jsou po této operaci data zcela jiná
- často se pojí s funkcí `summarise` (viz další slajd)

```
diamonds_grouped <- diamonds %>%  
  group_by(color, cut)
```

Výpočet charakteristik skupiny

- funkce summarise ale i summarize
- výpočet proměnných specificky pro každou skupinu vytvořenou pomocí funkce group_by

```
diamonds_grouped <- diamonds_grouped %>%  
  summarise(mean_price = mean(price),  
            median_carat = median(carat),  
            sum_price = sum(price),  
            count = n())
```

Propojení tabulek pomocí joinů

- klasické nástroje pro propojení tabulek pomocí klíčových atributů
- funkce `inner_join`, `left_join`, `right_join`, `full_join`, `semi_join`, `anti_join`
- důležitý parameter `by`, uvádí se bud' název proměnné, nebo v podobě `by = c("prom_tab_1" = "prom_tab_2")`

```
data("flights")
data("airlines")
joined_data <- flights %>% left_join(airlines, by = "carrier")
View(joined_data)
```

Další funkcionalita dplyr

- v helpu
- např. funkce distinct, if_else, sample, transmute
- příklady ve vinětách
- a další

Vstup a výstup dat

- sada funkcí ve základním R, ale nejsou příliš logické z hlediska názvů a parametrů
- vhodnější je využití balíčků `readr`, `readxl`, případně i `writexl`
- načítací funkce pak mají podobu `read_typsouboru`, zapisovací `write_typsouboru`
- týká se hlavně tabelárních dat, ostatní typy dat se načítají přes specializované knihovny

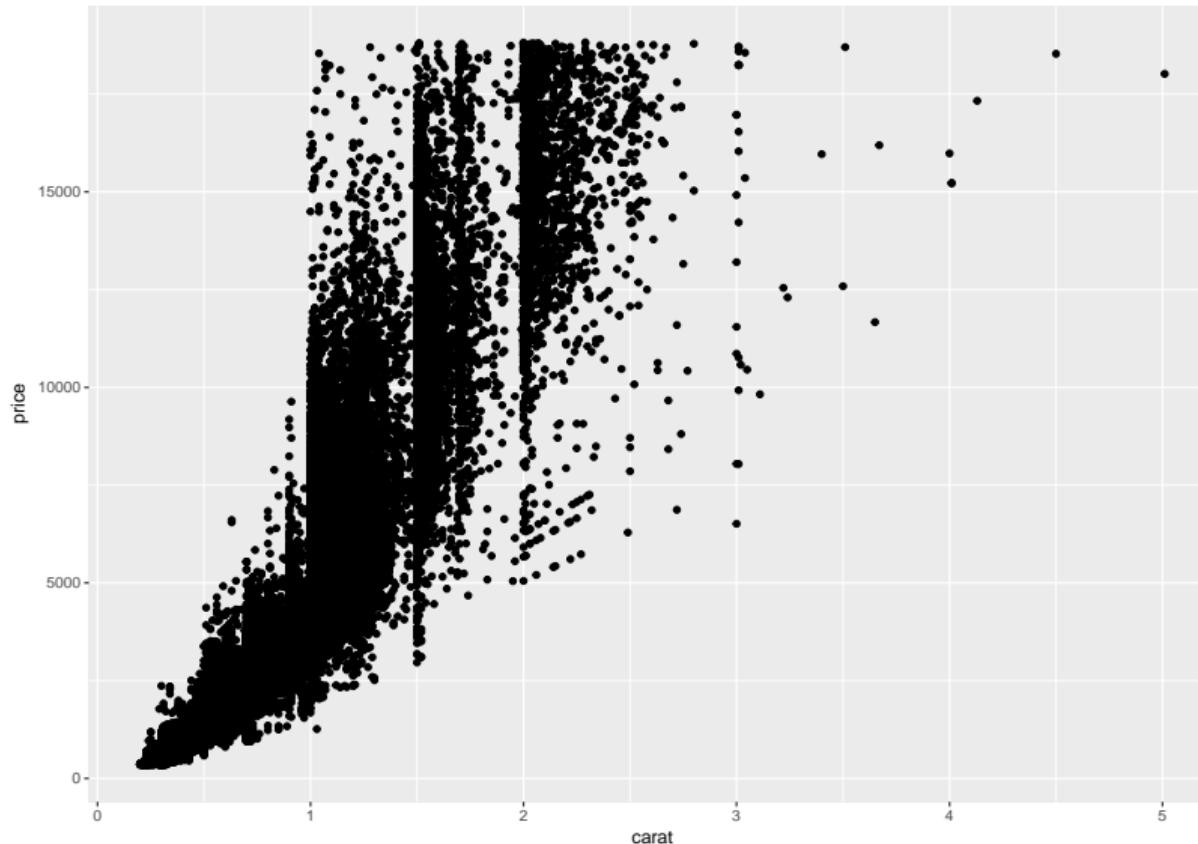
Tvorba grafů a dalších grafických výstupů

- teoreticky tuto funkcionality umí i základní R, pouze s balíčkem *grid*
- poměrně dost komplikované a nepraktické pro většinu reálných situací
- výrazně vylepšeno balíčkem *ggplot2* (Wickham, 2016), který přidává dost funkcionality a značně zjednodušuje širokou škálu nastavení
- umožňuje také poměrně jednoduše grafy stylovat
- rozšiřitelnost pomocí extenzí - <http://www.ggplot2-exts.org/gallery/>

Ukázka základního grafu

```
ggplot(diamonds_new, aes(carat, price)) +  
  geom_point()
```

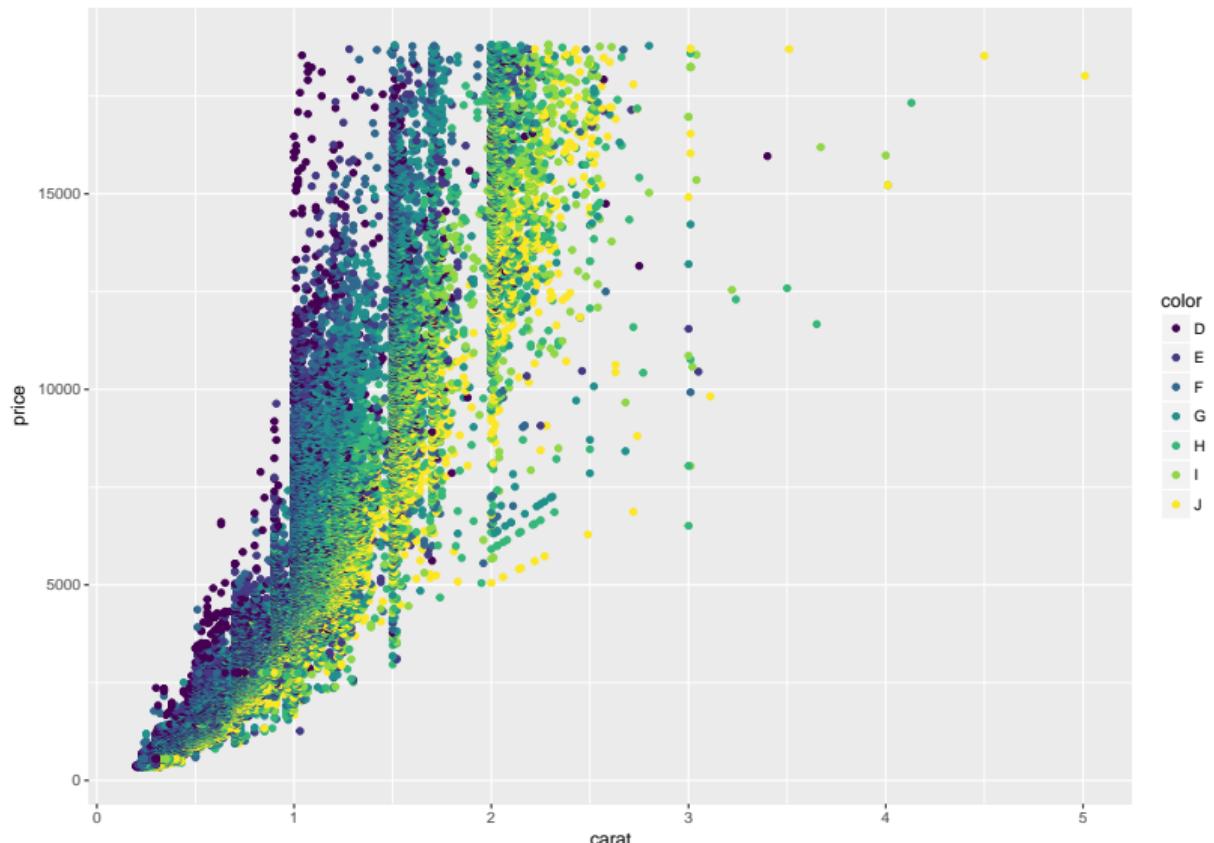
Ukázka základního grafu



Přidání barvy

```
ggplot(diamonds_new, aes(carat, price, color = color)) +  
  geom_point()
```

Přidání barvy

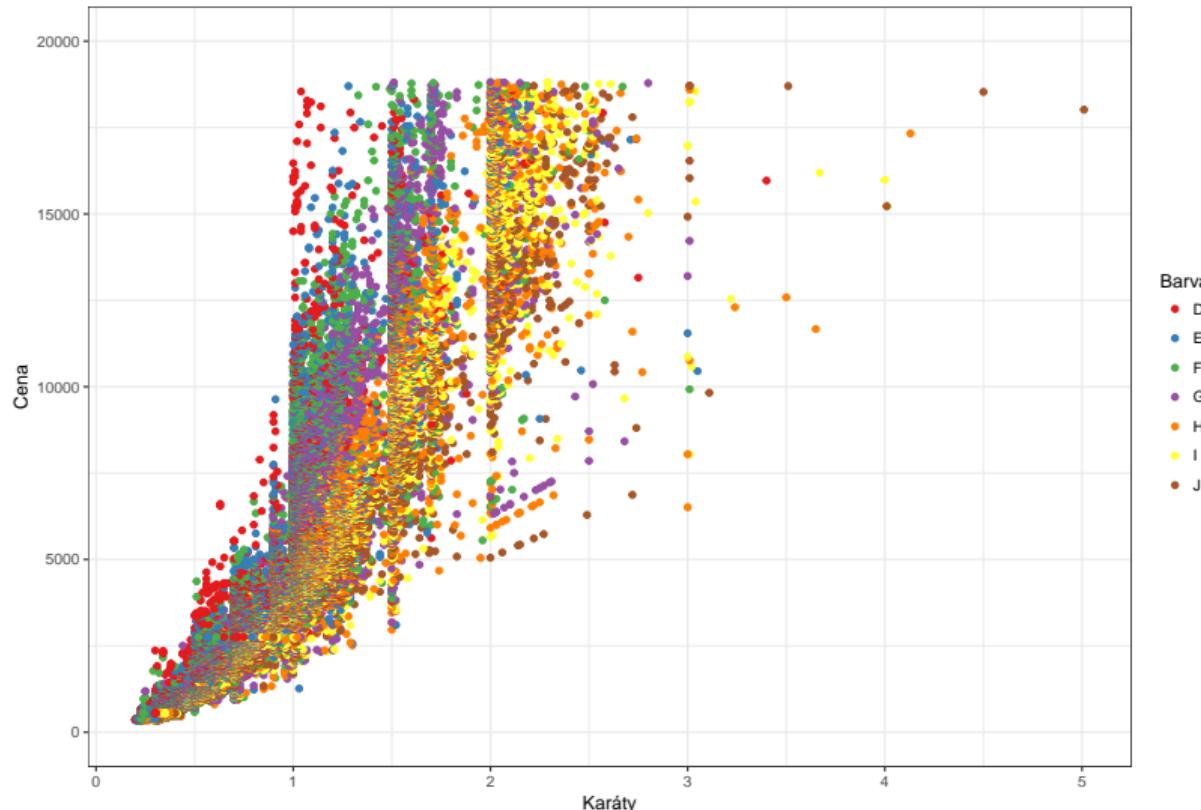


Doladění grafu

```
ggplot(diamonds, aes(carat, price, color = color)) +
  geom_point() +
  scale_color_brewer(palette = "Set1") +
  scale_y_continuous(breaks = seq(0, 20000, by = 5000),
                     limits = c(0, 20000)) +
  labs(x = "Karáty", y = "Cena", color = "Barva",
       title = "Porovnání ceny diamantů podle karátů a barvy") +
  theme_bw()
```

Doladění grafu

Porovnání ceny diamantu podle karátu a barvy



Další položky ggplot2

- různé geometrie geom_název
- možnosti stylovaní funcí theme
- uložení aktuálního grafu
- vložení grafu do proměnné a jeho uložení funkcí ggsave

```
ggsave("graf.pdf")
```

Literatura I

- Bivand, R. S. (2000) Using the R statistical data analysis language on GRASS 5.0 GIS database files. *Computers and Geosciences*, 26(9-10), pp. 1043–1052. doi:10.1016/S0098-3004(00)00057-1.
- Bivand, R. S. (2003) Approaches to Classes for Spatial Data in R. In: Hornik, K., Leisch, F., and Zeileis, A. (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), March 20-22, Vienna, Austria*. Vienna, Austria.
- Dhar, V. (2013) Data science and prediction. *Communications of the ACM*, 56(12), p. 64.
- Hijmans, R. J. (2017) *raster: Geographic Data Analysis and Modeling*. R package version 2.6-7.
URL: <https://CRAN.R-project.org/package=raster>
- Ihaka, R. and Gentleman, R. (1996) R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5(3), pp. 299–314. doi:10.1080/10618600.1996.10474713.
- Pebesma, E. (2017) *sf: Simple Features for R*. R package version 0.5-5.
URL: <https://CRAN.R-project.org/package=sf>
- Pebesma, E. and Bivand, R. (2005) Classes and methods for spatial data in R. *R-NEWS*, 5(2), pp. 9–13.
- Wickham, H. (2016) *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4.
URL: <http://ggplot2.org>
- Wickham, H., Francois, R., Henry, L., and Müller, K. (2017) *dplyr: A Grammar of Data Manipulation*. R package version 0.7.4.
URL: <https://CRAN.R-project.org/package=dplyr>

Dotazy?
Děkuji za pozornost.