



METODY VYHODNOCOVÁNÍ VODOHOSPODÁŘSKÝCH DAT

Martin Hanel, Adam Vizina

verze 1.0 (červenec 2014)

Skripta vznikla za finanční podpory projektu OP-Praha Adaptabilita CZ.2.17/3.1.00/36149 „Modernizace výuky udržitelného hospodaření s vodou a půdou v rámci rozvíjejících se oborů bakalářského studia“



EVROPSKÝ SOCIÁLNÍ FOND
PRAHA & EU: INVESTUJEME DO VAŠÍ
BUDOUCNOSTI

Recenzoval

Ing. Stanislav Horáček, Ph.D.
Ing. Roman Kožín

Obsah

1	Úvod	5
2	Úvod do R	7
2.1	Instalace a základní používání	7
2.2	Vektory	11
2.3	Matice a pole	16
2.4	Faktory a tabulky četnosti	18
2.5	data.frame	22
2.6	Typ, mód a třída objektů	24
2.7	Funkce a řídicí struktury	27
2.8	Základní grafika	30
2.9	Vybraná pokročilá téma	32
2.9.1	data.table	32
2.9.2	dplyr	33
2.9.3	DateTime třídy	33
2.9.4	ggplot2	34
2.9.5	GIS v R	34
3	Náhodné veličiny	35
3.1	Distribuční a kvantilová funkce, hustota pravděpodobnosti	35
3.2	Základní charakteristiky rozdělení pravděpodobnosti	37
3.3	Práce s rozděleními v R	43
3.4	Vybraná rozdělení v R	48
4	Popisná statistika	53
4.1	Empirická distribuční funkce a čára překročení	53
4.2	Výběrové kvantily	56
4.3	Výběrové momenty	58
4.4	Míry polohy výběru	58
4.5	Míry variability	60
4.6	Míry tvaru	61
4.7	Histogram	61
4.8	Krabicový graf - <i>Boxplot</i>	64
5	Vodohospodářská data a jejich vizualizace	67
5.1	Typy dat	67
5.1.1	Atmosférické proměnné	68
5.1.2	Hydrologická data	69
5.2	Vizualizace a typy grafů	70
5.2.1	Vizualizace dat	70
5.2.2	Typy grafů	70
5.2.3	Typy základních grafů a jejich tvorba v R	72
5.2.4	Tvorba základních grafů	75
5.2.5	Způsob zobrazení a zavádějící grafy	76
5.2.6	R a <i>GIS</i>	78
6	Zpracování časových řad	79
6.1	Úvod do časových řad	79
6.2	Dekompozice časové řady	80
6.2.1	Aproximace trendu matematickými funkcemi	81

6.2.2	Sezónní složka	82
6.3	Klouzavý průměr	83
6.4	Korelace a regrese	84
6.4.1	Kovariance a korelace	84
6.4.2	Autokorelace	85
6.4.3	Régresní analýza	86
6.5	Metoda nejmenších čtverců	89
7	Odhad parametrů a testování hypotéz	91
7.1	Odhad parametrů a jeho vlastnosti	91
7.2	Maximálně věrohodný odhad	96
7.3	Testování hypotéz	98
8	Volba modelu a vyhodnocení shody	105
8.1	Data pro vyhodnocení	105
8.1.1	Binární události	105
8.1.2	Kontinuální veličiny	106
8.2	Vyhodnocení pravděpodobnostních modelů	108
9	Analýza extrémů	113
9.1	Definice extrémů	113
9.2	Teoretické modely extrémů	116
9.3	Gumbel plot	118
9.4	Extrémy v R	119
10	Hydroklimatické indexy	125
10.1	Klimatické indexy	125
10.2	Indexy pro hodnocení nedostatku vody	128
10.3	Sucho	128
10.3.1	Propagace sucha	128
10.4	Vybrané indexy pro hodnocení sucha	130
10.4.1	SPI	130
10.4.2	PDSI - Palmer Drought Severity Index	133
10.4.3	Nedostatkové objemy	135
10.4.4	Další často užívané indexy	135

1 Úvod

Toto skriptum slouží primárně jako studijní materiál k předmětu Metody vyhodnocování vodohospodářských dat (který je vyučován na Fakultě životního prostředí České zemědělské univerzity v Praze). Cílem je seznámit se se základy zpracování, prezentace a jednoduché průzkumové analýzy dat, zejména vodohospodářských. Důraz je kladen především na praktické využití prezentované látky, proto je součástí textu řada příkladů využívajících prostředí R.

Ačkoliv není možné se zcela vyhnout prezentaci statistické teorie, v žádném případě není cílem těchto skript nahrazovat učebnice statistiky, ale spíš její výuku doplnit a obohatit o zkušenosti získané prací s daty v oblasti hydrologie, meteorologie a klimatologie. Hlubší studium statistiky proto doporučujeme všem, které probíraná téma zaujmou, a především těm, kteří budou využívat aplikovanou statistiku v praxi. Jako základní doplňkové materiály je proto vhodné využívat některá z dostupných skript statistiky např. Puš (2011); Jarušková (2011).

První ucelená verze učebních materiálů vznikla v rámci podpory z Operačního programu Praha během zimního semestru 2013/2014. Tato podoba nicméně není finální a počítáme s průběžným doplňováním, rozšiřováním tématických okruhů a odstraňováním případných chyb, přinejmenším po dobu vyučování předmětu Metody vyhodnocování vodohospodářských dat. Verze, kterou právě čtete, tudíž nemusí být nejnovější a doporučujeme proto její aktuálnost ověřit. Jakékoli postřehy, informace o chybách či nejasnosti týkající se tohoto materiálu je možno zaslat prostřednictvím emailu hanel@fzp.czu.cz.

Text je strukturován následovně: Kapitola Úvod do R (kapitola 2) popisuje základy práce s daty ve statistickém prostředí R. Cílem je seznámení se základy efektivní práce s Rkem, jsou zmíněny základní datové struktury, práce s nimi, základní grafika a vybraná pokročilá téma zaměřená na efektivní práci s daty, vizualizaci dat pomocí balíku ggplot2, využití Rka jako nástroje pro jednoduché GIS analýzy atd.

Kapitola Náhodné veličiny (kapitola 3) stručně popisuje koncept náhodné veličiny ve statistice, definuje distribuční a kvantilovou funkci a rozdelení pravděpodobnosti. Dále jsou prezentována některá nejpoužívanější rozdelení. Důraz je kladen na práci s rozdeleními v Rku - tj. zjišťování teoretických pravděpodobností, kvantilů a hustoty a vizualizaci rozdelení.

V kapitole Popisná statistika (kapitola 4) prezentujeme základní ukazatele sloužící k charakterizaci vlastností výběru dat, zejména se věnujeme mírami polohy, variability a tvaru, dále vícečíselným charakteristikám a summarizaci rozdelení pomocí boxplotu či histogramu.

Kapitola 7 prezentuje stručně základní principy odhadu parametrů a testování hypotéz. Diskutovány jsou zejména vlastnosti odhadů a metody výpočtu intervalů spolehlivosti. Dále je nastíněn princip testování od formulace hypotézy přes tvorbu zamítacích pravidel a vyhodnocení testu. Vše je opět doprovázeno příklady.

Skripta uzavírá kapitola Analýza extrémů (kapitola 9) popisující možnosti definice extrémů, jejich teoretické rozdelení a odhad jejich parametrů.

Vysvětlivky

Na mnoha místech budeme používat R kód. Ten je vyznačen písmem s pevnou šířkou, např.

```
> log(10)
```

```
[1] 2.303
```

Symbol > je tzv. prompt, který indikuje vstup na konzoli - tj. co je za symbolem >, píšeme do konzole, zbytek je odpověď Rka. V případě, že nedokončíme příkaz, se tento symbol změní na + a Rko čeká, než příkaz dokončíme (např. doplníme chybějící závorku):

```
> log(  
+ 10  
+ )  
[1] 2.303
```

Součástí kódu mohou být i komentáře uvedené symbolem #, které se neprovádějí, pouze vypisují.

```
> # toto je komentář
```

V kódu se může vyskytovat několik dalších typů informací, zejména chyby (Error), např.

```
> log('ahoj')  
Error: non-numeric argument to mathematical function
```

- není možné počítat logaritmus textového řetězce, varování (Warning), např.

```
> log(-1)  
Warning: NaNs produced  
[1] NaN
```

- logaritmus není definován pro záporná čísla, Rko vrací symbol NaN, tj. *not a number* - není číslo, a zprávy (Message), např. při načtení balíku:

```
> require(data.table)
```

V případě chyby se další kód neprovádí, varování nás pouze upozorňuje, že něco neproběhlo standardním způsobem, nicméně kód se provede, zprávy nás informují např. o nahrání balíků apod. Ve všech případech je užitečné věnovat těmto informacím pozornost.

2 Úvod do R

2.1 Instalace a základní používání

Rko je programovací jazyk a prostředí pro statistické výpočty a grafiku. Rko mimo jiné

- umožňuje efektivní manipulaci a ukládání dat
- obsahuje sadu operátorů pro práci s datovými poli, zejména maticemi
- obsahuje velkou konzistentní sadu nástrojů pro pokročilou analýzu dat
- disponuje nástroji pro vizualizaci a prezentaci dat
- je jednoduchým a efektivním programovacím jazykem

Termín „prostředí“ pro statistické výpočty zdůrazňuje, že Rko je flexibilním, nicméně přesně definovaným systémem. Základní Rko je neustále rozšiřováno pomocí balíků (packages).

Rko pro Windows lze stáhnout z webových stránek <http://www.r-project.org> (v levém menu **CRAN → výběr serveru (např. Austria) → Download R for Windows → base → Download R X.XX for Windows**).

Základní distribuce Rka umožňuje přístup k Rku buď prostřednictvím konzole (zjednodušeně příkazové řádky) nebo v rámci jednoduchého uživatelského rozhraní R GUI, které mimo konzole obsahuje i základní nástroje pro editaci skriptu (programu), export grafických výstupů apod. Přestože prostřednictvím konzole máme přístup k veškeré funkcionality Rka, není práce s ní (interaktivní zadávání jednotlivých příkazů) vždy nejpohodlnější ani nejfektivnější. Například v situaci, kdy provádíme opakovaně sekvenci nějakých příkazů se nabízí tyto příkazy uložit do textového souboru (tzv. skriptu) a opakovaně jej spouštět, případně obměňovat. Skript můžeme editovat v jakémkoliv editoru textových souborů (včetně Notepadu, MS Wordu apod.), nicméně pro efektivní práci s kódem v jakémkoliv programovacím jazyku je vhodné, aby tento editor umožňoval zvýrazňování syntaxe (např. jinou barvou se zobrazují textové řetězce, čísla, rezervovaná slova atp.), případně doplňování kódu, správu přídavných balíků ad. Toto konzole ani základní R GUI neumožňuje. Z tohoto důvodu existuje pro Rko řada tzv. IDE (Integrated Development Environment), poskytující různou míru uživatelského komfortu. V současnosti nejpokročilejším IDE je pravděpodobně RStudio.

RStudio mimo jiné

- integruje konzoli s editorem kódu, prohlížečem návodů, grafických výstupů a systému souborů
- umožňuje efektivní práci s kódem (zvýrazňování syntaxe, doplňování kódu po stisku tabulátoru, systém klávesových zkratek)
- umožňuje spouštění celého skriptu, aktuálního řádku či výběru pomocí tlačítka nebo klávesových zkratek
- umožňuje prohlížení objektů nahraných do paměti prostřednictvím Rka
- obsahuje nástroje pro integraci R kódů do prostředí LaTeX a Markdown (systémy pro efektivní tvorbu reportů, rozsáhlejších dokumentů a webových stránek)
- umožňuje tvorbu html prezentací
- poskytuje integrovanou podporu verzovacích systémů Git a Svn

Web RStudio se nachází na <http://www.rstudio.org>, instalační soubor najdete pod **Download RStudio → Download RStudio Desktop**. V linuxových distribucích je Rko v repozitářích (balík r-base), v případě RStudio je nutné jeho balík stáhnout z webu.

Seznámení s RStudiem

Po prvním spuštění je okno RStudio rozděleno na tři části:

- v levé části se nachází konzole pro psaní příkazů,
- v pravé horní části okno s přehledem dat (zatím prázdné) a historií příkazů,
- v pravé dolní části okno se správcem souborů, grafy, balíky a návodou.

Práce s Rkem spočívá v psaní kódu (v programovacím jazyku R), který je následně zpracováván vlastním „programem“ R (interpretom programovacího jazyka). Vložíme-li do konzole například

> [446+446](#)

[1] 892

Rko zadany řádek vyhodnotí (protože umí vyhodnocovat matematické výrazy jako kalkulačka) a vypíše výsledek.

ÚKOL 2.1 Vypočtěte v konzoli, kolik má rok minut.

ÚKOL 2.2 Vyzkoušejte, k čemu slouží v konzoli šipky nahoru a dolů a tabulátor.

ÚKOL 2.3 Kolik je $1/0$, $0/0$, $-1/0$?

Dostupné funkce a návod

Rko obsahuje řadu funkcí, další funkce je možno získat v doplňujících balících a rovněž je možné vytvářet vlastní funkce. Funkce je definována názvem, seznamem argumentů a tělem funkce. Pomocí názvu funkci voláme, názvem je tedy např. `mean`, `log`, `plot` apod. Argumenty funkce jsou zjednodušeně objekty, se kterými funkce poté pracuje. Tělo funkce definuje operace, které se provádějí. Argumenty a tělo funkce lze zjistit pomocí funkcí `args` a `body`, napsáním názvu funkce do konzole nebo nejlépe pomocí návodu. Tu je možno zobrazit pomocí funkce `help` nebo zkráceně voláním `?následovanám jménem funkce`, např. `?seq`, v RStudiu i pomocí klávesy F1 stisknuté na funkci, k níž hledáme návodu (F2 zobrazí tělo funkce). Návodu k určitému balíku je možno zobrazit pomocí `help(package=x)`. V tomto případě je `package` argumentem funkce `help`, kterému zadáváme hodnotu `x`, tj. název balíku, pro který chceme zobrazit návodu (např. pokud chceme zobrazit návodu k balíku `stats`, píšeme `help(package = "stats")`).

Návod má pevně danou strukturu. `Description` (popis) popisuje, co funkce dělá, `Usage` (využití) uvádí možné typy použití. Součástí této části návodu je i popis volání funkce včetně argumentů. Argumenty jsou v kulatých závorkách buď jen vyjmenovány, v tom případě nemají nastavené výchozí hodnoty a je nutné je specifikovat ve volání funkce, nebo jsou udány ve formě jméno argumentu = hodnota, v tom případě není-li nastavení argumentu součástí volání funkce použije se tato hodnota. Někdy může být argument uveden i ve formě `argument = vektor hodnot` udávající možné hodnoty, výchozí hodnotou je pak zpravidla první prvek vektoru. Součástí seznamu argumentů může být i speciální argument ..., který říká, že v principu je možné zadat další argumenty, které jsou v rámci funkce předány jiné volané funkci.

Například funkce `cor` má následující argumenty:

```
> args(cor)
function (x, y = NULL, use = "everything", method = c("pearson",
    "kendall", "spearman"))
NULL
```

přičemž

- x musí být zadáno (volání `cor()` vede k chybové hlášce)
- y má zadanou výchozí hodnotu (NULL) a zadávat se nemusí, tj. `cor(x = matrix(1:10))` projde bez chyby
- `use` má zadanou výchozí hodnotu ("everything") a zadávat se nemusí
- `method` má výchozí hodnotu "pearson" a další možné hodnoty jsou "kendall" a "spearman". V případě hodnot argumentů daných výčtem je zpravidla možné názvy zkracovat, tedy `cor(x = matrix(1:10), method = "spe")` je platným voláním a Rko vypočítá spearmanův korelační koeficient

Sekce **Arguments** (argumenty) jednotlivé argumenty popisuje, další popis funkce je uveden v sekci **Details**(podrobnosti). Funkce zpravidla vrací nějakou hodnotu/hodnoty (ale nemusí, některé funkce jsou např. volány pro kreslení grafů), tyto hodnoty jsou popsány v části **Value**(hodnota), **References**(odkazy) udává odkazy na literaturu (např. popis použité statistické metody) a **See Also**(viz i) nabízí podobná téma. Užitečné bývají také závěrečné příklady (část **Examples**(příklady)).

Funkce je možné volat několika způsoby:

- v případě, že funkce nemá žádné argumenty nebo v případě, že všechny argumenty mají v rámci definice funkce přiřazené výchozí hodnoty, je možno volat funkci bez argumentů, srov. např. `system.time()` a `seq()`
- pomocí nepojmenovaných argumentů - např. `seq(0, 1, .1)`. Argumenty pak musí být ve stejném pořadí, v jakém jsou definovány (a zobrazeny v návodě).
- pomocí pojmenovaných argumentů - např. srov. `seq(by = .1, to = 1, from = 0)` a `seq(.1, 1, 0)`

Proměnné

Proměnné umožňují přiřadit jméno určitému objektu, například číslu, a dál pak s ním symbolicky pracovat. V Rku vzniká proměnná automaticky, když dosud neznámému jménu přiřadíme hodnotu. Zadání proměnné

```
> ahoj
```

```
Error: object 'ahoj' not found
```

ohlásí chybu, protože proměnná s tímto názvem nebyla dosud vytvořena. Přiřazením

```
> ahoj = 135
```

se proměnná vytvoří – v RStudiu se zobrazí spolu se svou hodnotou v pravém horní části na záložce **Workspace**. Od té chvíle můžeme s proměnnou pracovat:

```
> ahoj^2
```

```
[1] 18225
```

Proměnné je třeba pojmenovávat znaky bez diakritiky a bez mezer. Místo mezer je možné použít podtržítka, pomlčky nebo tečky.

ÚKOL 2.4 Zjistěte, zda v Rku záleží na velikosti písmen v názvech proměnných.

ÚKOL 2.5 Znovu vypočtěte, kolik má rok minut, tentokrát ale vytvořte proměnné `dni_v_roce` a `minut_za_den` a použijte je při výpočtu.

Kromě operátoru = zajišťujícího přiřazení se v Rku často používá i operátor <- respektive ->. Jejich význam je v podstatě stejný a operátory jsou ve většině případů zaměnitelné. Výjimkou jsou argumenty funkce, kde = nastavuje hodnotu argumentu, kdežto <- zároveň vytváří proměnnou. Operátory = a <- přiřazují výrazu na levé straně hodnotu výrazu na pravé straně, jediným způsobem, jak přiřazovat naopak, je ->. Srovnej:

```
> a = 5
> b <- 10
> d <- a <- 2
> d = a = 2
> 5 -> e
> 5 = e

Error: invalid (do_set) left-hand side to assignment

> sin(x=10)
[1] -0.544

> x

Error: object 'x' not found

> sin(x<-10)
[1] -0.544

> x
[1] 10
```

Seznam všech proměnných je možné vypsat příkazem `ls()`. Stejný seznam se objevuje v okně Workspace RStudio. Proměnnou je možné smazat použitím `rm(ahoj)`, všechny proměnné najednou se smažou pomocí

```
> rm(list = ls())
```

případně v RStudiu ikonou **Clear All**.

Vytvoření a spuštění skriptu

Pro opakování použití není praktické zadávat kód do konzole, ale ukládat ho do souboru (skriptu) a ten pak spouštět. V RStudiu vytvoříme nový skript volbou **File – New – R Script** (případně ikonou nebo klávesovou zkratkou). V levé horní části se objeví okno s textovým editorem, ve kterém se skript bude psát. Skript je třeba uložit (**File – Save**, ikona, **Ctrl+S**), je vhodné používat příponu **.r** nebo **.R**.

Nyní můžeme do editoru psát kód na libovolný počet řádků, aniž by se spouštěl.

ÚKOL 2.6 Zadejte do editoru předchozí příklad – výpočet počtu minut v roce pomocí proměnných.

Spustit kód uložený ve skriptu („poslat kód do konzole“) je možné více způsoby:

- **Ctrl+Enter** (ikona **Run**, **Code – Run line(s)**) spustí aktuální řádek nebo označený kód,
- **Ctrl+Alt+R** (**Code – Run region – Run all**) spustí celý skript,
- **Ctrl+Shift+S** (ikona **Source**, **Code – Source**) spustí celý skript pomocí funkce `source`,
- ... a další.

ÚKOL 2.7 Jaký je v RStudiu rozdíl mezi obyčejným spuštěním a spuštěním pomocí source?



Nastavení pracovního adresáře

Při načítání souborů do Rka je dobré znát aktuální pracovní adresář (*working directory*). Vzhledem k tomuto adresáři se pak udává relativní cesta k souborům (je také možné, ale méně pohodlné zadávat cestu absolutní).

V RStudiu lze pracovní adresář nastavit tak, že ve správci souborů (pravé dolní okno, záložka **Files**) vstoupíme do požadovaného adresáře a zvolíme ikonu **More – Set As Working Directory**. Tím jsme vykonali funkci

```
> setwd("cesta-k-nejakemu-adresari")
```

ÚKOL 2.8 Nastavte pracovní adresář do nějaké složky, např. flash disk.

ÚKOL 2.9 Řetězec znaků se zadává jako v tomto případě v uvozovkách. Je možné místo uvozovek použít apostrofy?



Pro zjištění aktuálního pracovního adresáře slouží funkce `getwd()` (tato funkce nemá žádné argumenty). Po opětovném spuštění programu je potřeba pracovní adresář nastavit znova.

2.2 Vektory

Konstrukce vektorů a operace s vektory

Hlavní síla Rka spočívá v možnosti pracovat s vektory (a maticemi a poli - viz Kapitolu 2.3). Vektorem myslíme uspořádanou n-tici prvků, např. vektor x o délce 4 může být definován jako $x = (1, 2, 5, 10)$. Vektor se vytváří funkcí `c()`, například

```
> c(1, 2, 5, 10)
```

```
[1] 1 2 5 10
```

nebo

```
> c('Standa', 'Adam', 'Petr')
```

```
[1] "Standa" "Adam" "Petr"
```

Pro souvislou řadu celých čísel lze použít dvojtečku (například `3:10`). Složitější číselné posloupnosti je možné vytvářet pomocí funkce `seq()`:

```
> seq(from = 1, to = 5)
```

```
[1] 1 2 3 4 5
```

ÚKOL 2.10 Zjistěte jaké má funkce `seq` argumenty.

ÚKOL 2.11 Zjistěte k čemu slouží argumenty `by` a `length.out`.



Vektory je možno opakovat pomocí funkce `rep()`.

ÚKOL 2.12 Pomocí funkce `seq()` vytvořte vektor `(0, .2, .4)` a z něj pomocí funkce `rep()` vektor `(0, 0, .2, .2, .4, .4)`. Výsledek uložte do proměnné (např. `vys`).

```
> a = c(0, .2, .4)
> vys = rep(a, each = 2)
> vys
[1] 0.0 0.0 0.2 0.2 0.4 0.4
```

Proměnná `vys` je vektor. V RStudiu je v okně **Workspace** u této proměnné uveden datový typ (`numeric`, více o datových typech viz kapitolu 2.6) a délka, po klepnutí na tento údaj se zobrazí okno s hodnotami vektoru. S vektory je možné provádět stejné operace jako s čísly. Tedy např.

```
> y=vys*2
> y
[1] 0.0 0.0 0.4 0.4 0.8 0.8
> vys/y
[1] NaN NaN 0.5 0.5 0.5 0.5
```

V Rku je možné v rámci vektoru či proměnné indikovat chybějící hodnoty (tj. například chybějící měření z daného dne) pomocí konstanty `NA` (více o speciálních hodnotách viz kapitolu 2.6). Tedy např. s vektorem `x = c(1, NA, 5:10)` můžeme standardně počítat (např. `x*2`).

K prvkům vektoru se přistupuje pomocí hranatých závorek: `y[3]` vrátí třetí prvek vektoru (prvek s indexem 3) – indexování v Rku totiž začíná číslem 1 (v jiných programovacích jazycích tomu může být jinak). Dalšími možnostmi indexování se budeme ještě zabývat.

ÚKOL 2.13 Vypište najednou první tři a pátý prvek vektoru `y`.

Zvláštním případem indexování je, když v indexu použijeme záporné číslo. Potom se vrátí všechny prvky vektoru kromě těch, které byly jako záporné číslo uvedeny.

ÚKOL 2.14 Vypište vektor `y` bez první a poslední hodnoty.

Datum

Datum v Rku (proměnná typu `Date`) se ve výchozím nastavení zobrazuje ve formátu `YYYY-MM-DD`. Vytvoříme ho například jako

```
> d = as.Date("2012-09-29")
```

Sekvence (vektory) datumů lze v Rku vytvářet pomocí funkce `seq(from, to, by)`, kde `from` a `to` jsou datumy a `by` je časový krok ('day', 'week', 'month', 'year').

ÚKOL 2.15 Pomocí sekvence datumů zjistěte, kolik dní/týdnů uplynulo od vašeho narození. Délku vektoru zjistíte pomocí funkce `length()`

Datum je možno formátovat pomocí příkazu `format()` (který vrací textový řetězec):

```
> d  
[1] "2012-09-29"  
> format(d, '%Y')  
[1] "2012"  
> format(d, '%m')  
[1] "09"  
> format(d, '%d')  
[1] "29"
```

více viz `?format.POSIXlt`. Formátovací znaky je možno kombinovat:

```
> format(d, '%Y-%m')  
[1] "2012-09"  
> format(d, '%y/%m/%d')  
[1] "12/09/29"
```

ÚKOL 2.16 Zformátujte datum do tvaru **DD.MM.YYYY**.

Logické vektory

Logická proměnná (typ `logical`) může nabývat pouze dvou hodnot: pravda (`TRUE`) a nepravda (`FALSE`). Logické proměnné jsou vraceny z vyhodnocování výrazů, například:

```
> 65 < 24  
[1] FALSE  
> yyyy = 24 == 24  
> yyyy  
[1] TRUE
```

Dvojité rovnítko `==` není překlep, jde o operátor porovnávání (podobně jako `>`, `<`), který není zaměnitelný s operátorem přiřazení (jednoduché rovnítko `=`). Existuje i operátor `!=`, tj. „není rovno“. Logické vektory je možné jednoduše vytvořit pomocí operátorů porovnání `<`, `>`, `==`, `>=`, `<=` a `%in%`. Porovnávám-li víceprvkový vektor s jednoprvkovým, probíhá porovnávání s každým prvkem víceprvkového vektoru, tj.

```
> 1:10 < 5  
[1] TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE
```

V případě dvou víceprvkových vektorů probíhá porovnání po prvcích, případně je kratší z vektorů opakován. Pokud není délka delšího vektoru násobkem délky vektoru kratšího, vypíše se varování.

```

> 1:10 < 14:5
[1] TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE
> 1:6 < c(7, 5)
[1] TRUE TRUE TRUE TRUE TRUE FALSE
> 1:5 < c(7, 5)
Warning: longer object length is not a multiple of shorter object length
[1] TRUE TRUE TRUE TRUE TRUE

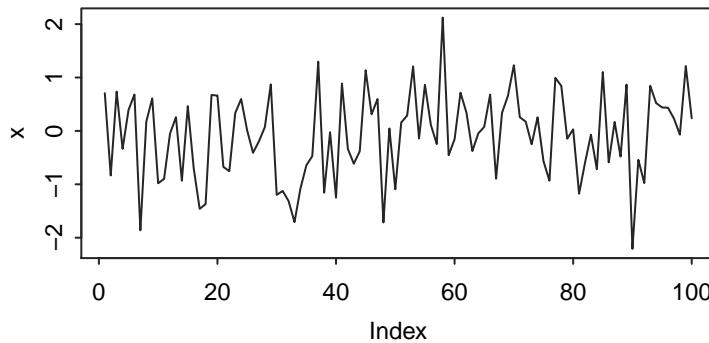
```

Vygenerujme např. řadu 100 čísel ze standardního normálního rozdělení a řadu vykresleme indeksovým grafem (znak # uvozuje komentář, cokoliv je za ním, je Rkem ignorováno),

```

> x = rnorm(100)
> plot(x, type='l') # 'l' jako v 'line', nikoliv číslo 1.

```



Obr. 2.1: Graf zobrazující 100 náhodných čísel.

pak například $x > 0$ vrátí logický vektor ukazující, pro která x platí, že $x > 0$. Použijeme-li tento vektor jako index, dostaneme všechny hodnoty splňující zvolenou podmínu:

```

> x[x>0]
[1] 0.70634 0.73612 0.39180 0.68039 0.17468 0.60821 0.25741 0.46443
[9] 0.67256 0.66196 0.33487 0.59720 0.01587 0.07299 0.87099 1.29654
[17] 0.88704 1.13564 0.31154 0.59590 0.04584 0.15749 0.28637 1.20800
[25] 0.86379 0.11194 2.12150 0.71637 0.34159 0.07573 0.68075 0.33769
[33] 0.65757 1.23013 0.25676 0.17300 0.25433 0.99365 0.84217 0.02876
[41] 1.10294 0.16789 0.86700 0.84465 0.52127 0.43917 0.43344 0.23244
[49] 1.21390 0.23747

```

Logické proměnné/vektory mohou figurovat i v přiřazení, tedy $x[x > 0] = 0$ dosadí nulu za všechna čísla vyšší než 0, podobně $x[x > 0] = 1/x[x > 0]$ dosadí za všechna kladná x jejich převrácenou hodnotu. Pokud chceme z logického vektoru získat vektor indexů prvků splňujících podmínu, docílíme toho pomocí funkce `which`, např. tedy `which(x >= 0.1)`.

Podmínky lze kombinovat s využitím logických operátorů:

Tab. 2.1: Logické operátory.

operátor	význam
&	logický součin, obě podmínky musí platit zároveň, aby byl výsledek pravda
	logický součet, musí platit jedna z podmínek
!	negace, vrátí se pravda, pokud je výraz nepravda

Operátory & a | jsou binární (vkládají se mezi dva výrazy), ! je unární (vkládá se před výraz). Srovnej:

```
> x = rnorm(10)
> x
[1] 0.7163 -2.1689 0.3607 -1.0611 0.3487 -0.6604 -0.0511 1.1808
[9] 1.3689 -0.1069

> # není menší než 0
> !(x < 0)
[1] TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE TRUE FALSE

> # větší než 0 a menší než 1
> x > 0 & x < 1
[1] TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE

> # které prvky splňují podmíinku?
> which(x > 0 & x < 1)

[1] 1 3 5

> # a kolik jich je?
> length(which(x > 0 & x < 1))

[1] 3
```

ÚKOL 2.17 Vypište všechny hodnoty vektoru x, které jsou větší než -1, a zároveň jsou menší než 1.

ÚKOL 2.18 Vygenerujte 100 hodnot ze standardního normálního rozdělení, záporná čísla nahraďte výrazem NA a vykreslete.

Logické vektory lze summarizovat pomocí výrazů all - pro všechny prvky, any - pro aspoň jeden z prvků, např.

```
> # jsou všechna x větší než nula?
> all(x>0)
[1] FALSE

> # jsou nějaká x větší než nula?
> any(x>0)
[1] TRUE
```

Indexování

V Rku je možné přistupovat k jednotlivým prvkům vektoru různými způsoby:

- číselnými indexy - např. x[5] nebo x[5:10]. V hranatých závorkách může být i proměnná nebo přímo výstup funkce - např. x[seq(1, 10, by = 2)]

- pomocí logických vektorů - např. `x[x > 0]`
- pomocí názvů

Vektor je možné pojmenovat pomocí funkce `names()`, tedy např.

```
> a=c(5,3)
> names(a) = c('jan','olin')
> a
jan olin
5     3
```

Pomocí `a['jan']` přitupujeme k jednotlivým položkám.

2.3 Matice a pole

Matice jsou dvourozměrná číselná pole. Vytvořit je můžeme např. pomocí funkce `matrix` - vytvoří matici z vektoru, `rbind` - spoj řádky nebo `cbind` - spoj sloupce. Rozměry (dimenze) matice zjišťujeme příkazem `dim`, příkaz `length` vrací celkový počet prvků.

```
> a = matrix(1:9, nrow=3)
> a
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

> b = matrix(1:9/10, nrow=3, byrow=TRUE, dimnames=list(radek=c('a','b','c'), sloupec = 1:3))
> b
      sloupec
radek   1   2   3
      a 0.1 0.2 0.3
      b 0.4 0.5 0.6
      c 0.7 0.8 0.9

> x = rnorm(5)
> y = rnorm(5)
> X = cbind(x, y)
> X
      x         y
[1,] -0.4755  0.4562
[2,]  0.7801  0.6852
[3,]  1.1491  0.8980
[4,]  0.1224  0.9909
[5,]  0.5458 -1.8337

> dim(X)
[1] 5 2

> Y = rbind(x, y)
> Y
      [,1] [,2] [,3] [,4] [,5]
x -0.4755  0.7801 1.149  0.1224  0.5458
y  0.4562  0.6852 0.898  0.9909 -1.8337

> dim(Y)
```

```
[1] 2 5
```

K jednotlivým dimenzím přistupujeme stejně jako k vektorům, dimenze jsou odděleny čárkou. Srov.

```
> X[1:2]
[1] -0.4755 0.7801
> # X[1:2] odpovídá zápisu c(X)[1:2] - tj. matice je nejprve převedena po sloupcích na vektor
>
> X[1:2, ]
      x      y
[1,] -0.4755 0.4562
[2,]  0.7801 0.6852
> # vymezíme index některé z dimenzí, vrací výraz pro tuto dimenzi všechny prvky - v tomto případě všechny
>
> X[3, ]
      x      y
1.149 0.898
> X[, 'y']
[1]  0.4562 0.6852 0.8980 0.9909 -1.8337
> b['a',]
  1   2   3
0.1 0.2 0.3
```

Všimněte si, že Rko v případě použití rbind a cbind samo pojmenovalo první dimenzi matice a názvy je možno použít pro indexování.

ÚKOL 2.19 Vytvořte matici náhodných čísel X, která bude mít 3 řádky a 10 sloupců.

ÚKOL 2.20 Vyberte 1. sloupec, 5. sloupec, 5. až 8. sloupec matice X.

ÚKOL 2.21 Vyberte 1. řádek, první dva řádky matice X.

■

Pole

Vícerozměrná pole je možno vytvářet pomocí příkazu array analogicky jako matice. Tento příkaz má argumenty x - data tvorící pole (často NA, pokud tvoríme prázdné pole, které poté plníme výsledky), dim - dimenze pole a dimnames - názvy dimenzí. Tedy např.

```
> a = array(NA, dim = c(2, 2, 2), dimnames = list(D1 = c('a', 'b'), D2 = c('x', 'y'), D3 = c('0', '1')))
> a
, , D3 = 0

      D2
D1   x  y
a  NA NA
b  NA NA

, , D3 = 1
```

```

D2
D1   x  y
  a NA NA
  b NA NA

```

Operace s maticemi a polí

Operátory $+$, $-$, $*$, $/$ fungují v případě matic a polí po prvcích. Lze je navíc použít i v případě operací mezi maticí (polem) a vektorem nebo maticí (polem) a číslem, tedy např.

```

> a = matrix(1, nrow=3, ncol=3)
> b = matrix(1:9, ncol=3)
> a+b

```

```

 [,1] [,2] [,3]
[1,]    2    5    8
[2,]    3    6    9
[3,]    4    7   10

```

```
> d = 1:3
```

```
> a+d
```

```

 [,1] [,2] [,3]
[1,]    2    2    2
[2,]    3    3    3
[3,]    4    4    4

```

```
> a+1
```

```

 [,1] [,2] [,3]
[1,]    2    2    2
[2,]    2    2    2
[3,]    2    2    2

```

Ke kombinacím matic (polí) a vektorů je potřeba přistupovat obezřetně, jelikož Rko jednak vektory, je-li potřeba, opakuje (recykuje) a jednak postupuje po sloupcích. To nemusí být vždy zcela vhodné.

Pro matice existuje sada speciálních funkcí:

Tab. 2.2: Maticové operace.

funkce	význam
%*%	maticové násobení
solve	inverze matice
diag	přístup k diagonálním prvkům
t	transpozice matice
upper.tri	vrací logickou matici s TRUE nad hlavní diagonálou
lower.tri	vrací logickou matici s TRUE pod hlavní diagonálou
eigen	odhad vlastních čísel a vektorů

2.4 Faktory a tabulky četnosti

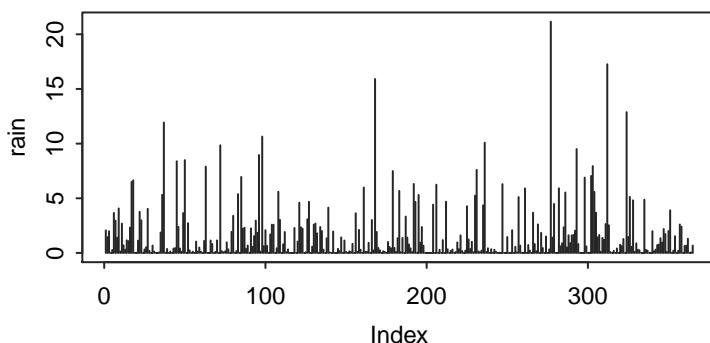
Faktor je (zpravidla) vektorový objekt, určující diskrétní klasifikaci jiného stejně dlouhého vektoru. Často jde o označení příslušnosti k určité třídě dat (např. pohlaví, měsíc v roce). Více informací o kategorických datech je uvedeno v kapitole ??.

Rko umožňuje provádět řadu operací pomocí faktorů, např. počítat skupinové průměry (obecně jakékoliv charakteristiky), vykreslovat data dle příslušnosti k jednotlivým skupinám apod. Faktor se tvoří funkcí `factor`, např. `fct = factor(c("muž", "žena", "žena", "muž", "žena"))`. Součástí faktoru jsou možné úrovně `levels`. Srovnej

```
> fct = factor(c("muž", "žena", "žena", "muž", "žena"))
> fct
[1] muž žena žena muž žena
Levels: muž žena
> levels(fct)
[1] "muž" "žena"
> levels(fct) = c('X','Y')
> fct
[1] X Y Y X Y
Levels: X Y
```

Pro další ilustraci vygenerujeme data z gama rozdělení a budeme je pokládat např. za roční měření srážek.

```
> dtm = seq(as.Date('2013-01-01'), as.Date('2013-12-31'), by = 'day')
> rain = 6 * rgamma(365, shape=.3)
> rain[rain<.05] = 0 # nastavíme malé srážky = 0
> plot(rain, type='h')
```



```
> sum(rain)
[1] 598.5
```

Faktory se často vytváří pomocí podmínek (např. `rain==0`). Vektor `rain==0` má stejnou délku jako vektor `rain` a rozděluje jednotlivé prvky na ty, pro které platí že nepršelo, tj. `rain==0` a ty pro které platí, že pršelo, tj. `rain!=0`.

Rko umožňuje pohodlnou tvorbu tabulek četnosti. Například můžeme rychle summarizovat, kolik dní pršelo a kolik ne:

```
> table(rain==0)
```

FALSE	TRUE
272	93

Pokud chceme data dále zkoumat, je možné psát

```
> # kolik srážek je vyšších než 10 mm?  
> table(rain>10)
```

```
FALSE TRUE  
358 7
```

```
> # kolik je to procent dní?  
> table(rain>0)/length(rain)*100
```

```
FALSE TRUE  
25.48 74.52
```

Dalším možným způsobem tvorby faktorů je diskretizace spojitéch proměnných pomocí funkce `cut`. Ta probíhá tak, že na nějakém spojitém intervalu definujeme kategorie (zpravidla malý počet) a vyhodnocujeme, kolik prvků náleží do jednotlivých kategorií. Kategorie lze pomocí funkce `cut` definovat dvěma způsoby:

```
> # počet tříd  
> kategorie = cut(rain, breaks = 5)  
> table(kategorie)  
  
kategorie  
(-0.0212,4.23] (4.23,8.46] (8.46,12.7] (12.7,16.9] (16.9,21.2]  
318 36 7 2 2  
  
> # přesně zadané hranice intervalů  
> kategorie = cut(rain, breaks = c(0, 2, 5, 10, 50))  
> table(kategorie)  
  
kategorie  
(0,2] (2,5] (5,10] (10,50]  
177 58 30 7
```

ÚKOL 2.22 Proč se vyskytují v zápisu závorky dvojitého typu? - tj. [, (

↳

Faktory je možné v rámci příkazu `table` kombinovat - tj. vytvářet kontingenční tabulky. Např.

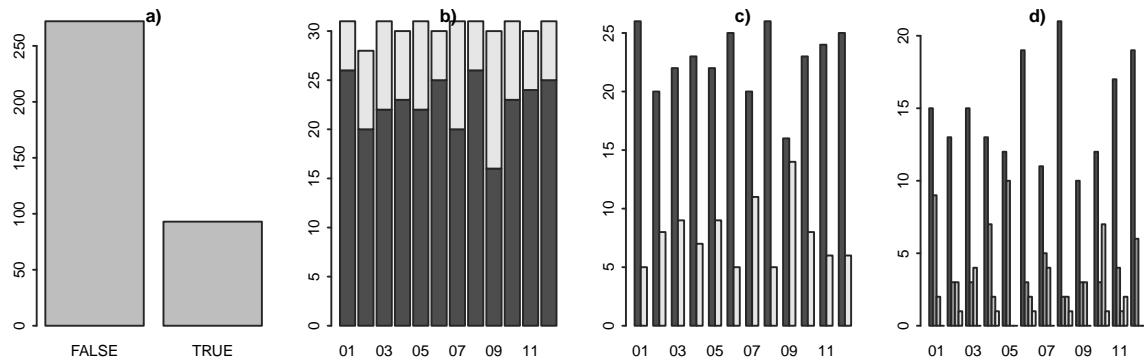
```
> kategorie = rain == 0  
> table(kategorie, format(dtm, '%m'))  
  
kategorie 01 02 03 04 05 06 07 08 09 10 11 12  
FALSE 26 20 22 23 22 25 20 26 16 23 24 25  
TRUE 5 8 9 7 9 5 11 5 14 8 6 6
```

Tabulky je možno zobrazit pomocí příkazu `barplot`. Tedy např.

```

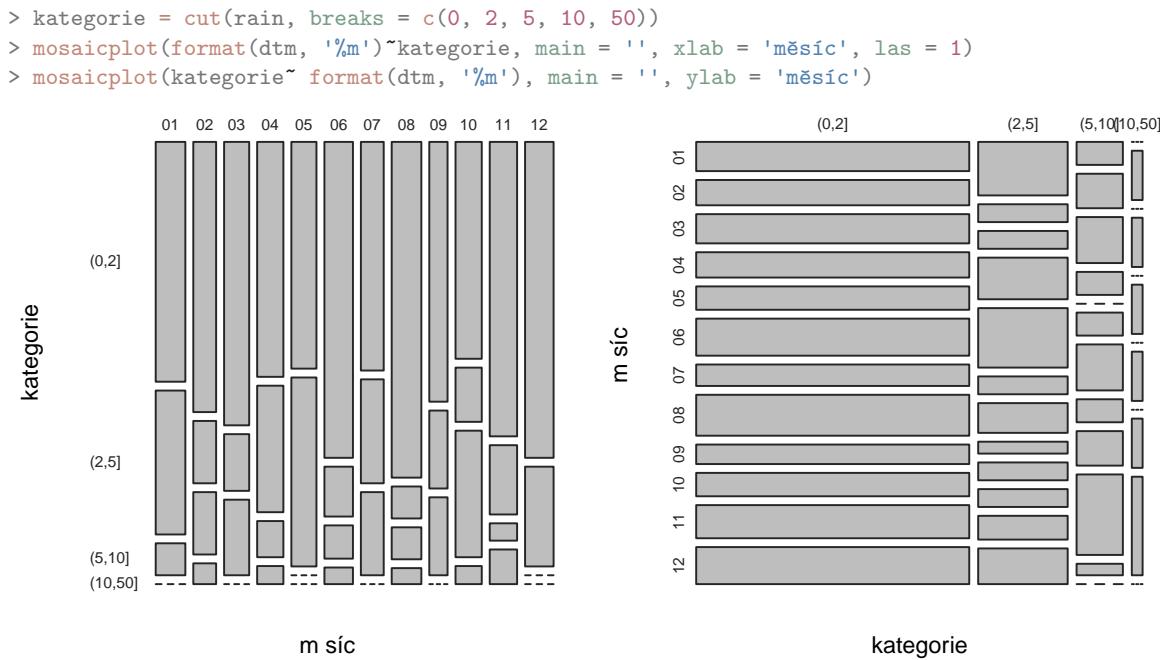
> kategorie = rain == 0
>
> # a)
> tab = table(kategorie)
> barplot(tab, main='a')
>
> # b)
> tab = table(kategorie, format(dtm, '%m'))
> barplot(tab, beside=FALSE, main='b')
>
> # c)
> barplot(tab, beside=TRUE, main='c')
>
> # d)
> kategorie = cut(rain, breaks = c(0, 2, 5, 10, 50))
> tab = table(kategorie, format(dtm, '%m'))
> barplot(tab, beside=TRUE, main='d')

```



Obr. 2.2: Vizualizace tabulek.

Kontingenční tabulky (tj. tabulky vztahů více proměnných) je možno zobrazit pomocí funkce `mosaicplot`.



Obr. 2.3: Vizualizace kontingenčních tabulek.

V grafu vlevo udává výška jednotlivých obdélníků počet událostí spadajících do jednotlivých kategorií. Šířka obdélníků odpovídá počtu událostí v jednotlivých měsících. Symbol --- vyjadřuje, že pro danou kombinaci proměnných nejsou žádné výskytu. Pro graf vpravo udává šířka obdélníků počet událostí v jednotlivých kategoriích, výška jednotlivých obdélníků odpovídá poměru zastoupení jednotlivých kategorií proměnné rain v jednotlivých měsících.

2.5 data.frame

Data.frame je zpravidla dvourozměrný objekt, zjednodušeně sada stejně dlouhých vektorů, které mohou být různého typu. Tvorba probíhá často pomocí příkazu `data.frame`, např. vytvořme fiktivní data o ranním jídelníčku smyšlené osoby:

```

> den = rep(c('pondělí', 'úterý', 'středa', 'čtvrtok', 'pátek'), length=50)
> snidane = sample(c('jogurt', 'banán', 'párek'), length(den), replace=TRUE)
> D = data.frame(DEN = den, JIDLO = snidane)

```

Data.frame je asi nejtypičtější datovou třídou používanou pro environmentální data. Data frame můžeme vyspat, případně přehledněji zobrazit pomocí `str(D)` (struktura objektu) – za znaky dolaru \$ uvidíme názvy jednotlivých vektorů (DEN, JIDLO atd.), jejich typ a několik počátečních hodnot. V případě rozsáhlejších data.frames (obecně jakýchkoliv objektů) je možno zobrazit pouze začátek pomocí funkce `head`, případně konec pomocí funkce `tail`.

Indexovat sloupce (proměnné) můžeme několika způsoby: pomocí číselných indexů podobně jako matice nebo pomocí názvů - srov.

```

> D[, 1]
> D[, 'DEN']
> D$DEN

```

Ke sloupcům lze přistupovat také pomocí jejich názvů a operátoru \$, například D\$DEN. Ekvivalentně je možno k sloupcům přistupovat pomocí dvojité hranaté závorky, tedy D[[1]] nebo D[['DEN']]. Sloupce lze odstranit z data framu pomocí D[[1]] = NULL. Jména sloupců (veličin) lze změnit pomocí funkce names. Jména řádků lze nastavit pomocí funkce rownames.

Srovnej

```
> names(D)
[1] "DEN"    "JIDLO"
> names(D)[2] = 'POKRM'
> names(D)
[1] "DEN"    "POKRM"
> D$x = 1
> head(D)
      DEN POKRM X
1 pondělí jogurt 1
2 úterý banán 1
3 středa párek 1
4 čtvrtek banán 1
5 pátek párek 1
6 pondělí jogurt 1
> D$x = NULL
> head(D)
      DEN POKRM
1 pondělí jogurt
2 úterý banán
3 středa párek
4 čtvrtek banán
5 pátek párek
6 pondělí jogurt
```

Řádky můžeme vybírat opět číselným indexem, názvem (pokud existuje) nebo pomocí logických vektorů. Tedy

```
> D[1:3,]
      DEN POKRM
1 pondělí jogurt
2 úterý banán
3 středa párek
> D[D$DEN=='pondělí',]
      DEN POKRM
1 pondělí jogurt
6 pondělí jogurt
11 pondělí banán
16 pondělí banán
21 pondělí banán
26 pondělí banán
31 pondělí párek
36 pondělí banán
41 pondělí banán
46 pondělí jogurt
```

ÚKOL 2.23 Z data.frame D zjistěte nejoblíbenější pondělní jídlo.

ÚKOL 2.24 Kolikrát byl na snídani banán a kolikrát párek?

+

2.6 Typ, mód a třída objektů

Objekty v Rku mají různé vlastnosti, atributy. Některé z nich mají všechny objekty. Jedná se o délku (zjišťujeme funkcí `length`), typ, mód a třídy. Objekty z hlediska interpretace Rkem mohou mít několik typů a módů. Význam obou termínů je podobný, přičemž mód objektu je obecnější. Mód objektu (zjišťujeme funkcí `storage.mode` případně `mode`, přičemž výstupy obou funkcí se mohou v některých případech lišit) v podstatě říká, jak je objekt uložen v paměti. Mezi základní módy objektů patří

logical	logická proměnná
integer	celé číslo
double	reálné číslo
complex	komplexní číslo
character	znakový řetězec
raw	bitová reprezentace objektů
list	seznam různých objektů
NULL	pro neexistující objekty
function	funkce
expression	výraz
environment	prostředí (např. ve kterém se výrazy vyhodnocují)

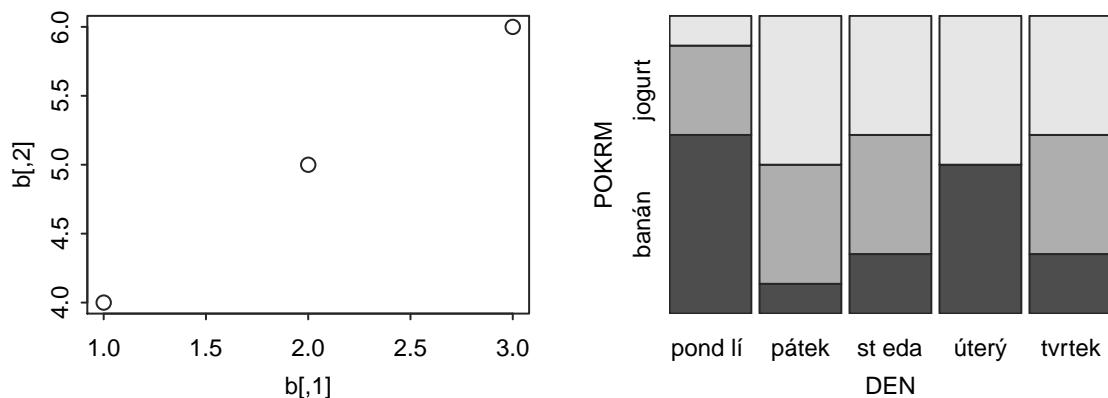
Funkce `mode` dává v podstatě stejné výsledky s tím, že např. `storage.mode` 'integer' a 'double' jsou oba označeny jako 'numeric'.

Typ objektu (zjišťujeme funkcí `typeof`) je v podstatě shodný se `storage.mode`m s tím, že mód `function` zahrnuje typy `closure` - obecná funkce, `builtin` - základní vestavěná funkce a `special` základní operátory (např. +, -).

Nezávisle na tom, jak jsou objekty reprezentovány v počítači, mohou mít objekty různé třídy (zjišťujeme funkcií `class`). Funkce, do kterých objekty vstupují se pak mohou chovat různě podle toho, jaké třídy je objekt, který do nich vstupuje. Příkladem může být funkce `length`, zjišťující délku objektu, nebo funkce `plot`. Srov.

```
> a = 1:5
> b = list(1:5)
>
> class(a)
[1] "integer"
> length(a)
[1] 5
> class(b)
[1] "list"
> length(b)
```

```
[1] 1
> b = matrix(1:6, ncol=2)
> plot(b)
> plot(D) # D je data.frame z kapitoly 2.5
```



Obr. 2.4: Ukázka funkce plot pro třídu matrix a data.frame.

Porovnání módů, typů a tříd objektů přináší následující tabulka.

Tab. 2.3: Módy, typy a třídy některých objektů.

objekt	storage.mode	mode	typeof	class
1	double	numeric	double	numeric
1L	integer	numeric	integer	integer
1:3	integer	numeric	integer	integer
factor("peklo")	integer	numeric	integer	factor
matrix(1:3)	integer	numeric	integer	matrix
matrix(1:3 - 0.5)	double	numeric	double	matrix
complex(real=10,imaginary = 3)	complex	complex	complex	complex
"ahoj"	character	character	character	character
sum	function	function	builtin	function
mean	function	function	closure	function
NULL	NULL	NULL	NULL	NULL
Inf	double	numeric	double	numeric
NA	logical	logical	logical	logical
NA_integer_	integer	numeric	integer	integer
NA_real_	double	numeric	double	numeric
NA_character_	character	character	character	character
NA_complex_	complex	complex	complex	complex

Pro převod mezi datovými typy slouží funkce začínající výrazem `as.` následovaným názvem třídy (případně `storage.mode`), na který chceme převádět:

```
> q = as.character(10)
> w = as.numeric(q)
> class(q)
```

```
[1] "character"
> class(w)
[1] "numeric"
> q * 2
Error: non-numeric argument to binary operator
> w * 2
[1] 20
```

V tomto příkladě mají proměnné `q` i `w` hodnotu 10, v prvním případě se však jedná o řetězec znaků, kdežto druhá proměnná je číslo.

Speciální hodnoty

V Tabulce 2.3 jsou uvedeny některé ze speciálních hodnot, které jsou v Rku k dispozici. Máme na mysli zejména `Inf`, `NULL`, `NA` a `NaN`. Přičemž

<code>Inf</code> , <code>-Inf</code>	reprezentují nekonečno
<code>NULL</code>	je, řekněme, prázdná množina
<code>NA (not available)</code>	slouží k reprezentaci chybějících hodnot
<code>NaN (not a number)</code>	slouží k reprezenaci výsledků operací, které nejsou definovány (např. jsou-li mimo definiční obor).

Pro chybějící hodnoty je někdy vhodné přímo definovat typ (jinak je výchozím typem `logical`), tedy např. `NA_integer_`, `NA_double_`, `NA_character_` atd. Rko zná tyto hodnoty, aby nebylo nutné hlásit chyby (a končit kód) pro operace typu `1/0`, případně aby bylo možno pracovat i s daty obsahující chybějící hodnoty (což je obecně relativně častý případ). Příklady využívající tyto hodnoty jsou v následující ukázce

```
> 1/0
[1] Inf
> Inf * Inf
[1] Inf
> Inf * 0
[1] NaN
> a = c(1, 2.3, NA, Inf, NaN)
> a*1
[1] 1.0 2.3 NA Inf NaN
> a/Inf
[1] 0 0 NA NaN NaN
> b = NULL
> length(b)
[1] 0
> b*2
numeric(0)
```

```

> d = c(NULL, NULL)
> length(d)
[1] 0
> d
NULL
> e = c(NULL, NULL, NA, 1)
> length(e)
[1] 2
> e
[1] NA  1

```

V Rku je možné zjišťovat různé vlastnosti objektů pomocí funkcí typu `is.x`, kde `x` je vlastnost, na kterou se dotazujeme, může to být třída objektu - např. `is.integer`, `is.character`, `is.numeric`, `is.factor` atd. nebo dotaz na speciální hodnotu, tedy `is.na`, `is.nan`, `is.finite`. Například

```

> is.integer(1)
[1] FALSE
> is.integer(1L)
[1] TRUE
> is.na(NA)
[1] TRUE
> is.na(NaN)
[1] TRUE
> is.nan(NA)
[1] FALSE
> is.finite(1/0)
[1] FALSE

```

2.7 Funkce a řídicí struktury

Funkce

Funkce je v Rku reprezentovaná jako proměnná typu `function`, vytváří se pomocí funkce `function`, jejímž argumentem je výpis argumentů tvořené funkce. Následuje tělo funkce, které je v případě složeného příkazu uzavřeno do složených závorek. Tedy např.

```

> ahoj = function() print('AHOJ!!')
> ahoj()
[1] "AHOJ!!"

```

definuje funkci `ahoj`, která nemá žádné argumenty a jediné, co dělá, je, že vypíše AHOJ!! Chceme-li funkci rozšířit, aby např. zahrnovala jméno pozdravené osoby, píšeme

```

> ahoj = function(jmeno){
+   print('AHOJ, ')
+   print(jmeno)

```

```

+         }
> ahoj('Honzo')

[1] "AHOJ,"
[1] "Honzo"

```

Udělali jsme v zásadě dvě úpravy - tělo funkce je uzavřeno ve složených závorkách a přidali jsme argument jméno, který dále funkce vypisuje. Tento argument nemá výchozí hodnotu a při volání ahoj() dostaneme chybu. Výchozí hodnotu specifikujeme, definujeme-li

```

> ahoj = function(jmeno = 'pane'){
+     print('AHOJ,')
+     print(jmeno)
+ }
> ahoj()

[1] "AHOJ,"
[1] "pane"

```

Hodnotu, kterou funkce vrací, je možno specifikovat v těle funkce pomocí funkce `return`. Pokud není vracená hodnota specifikovaná, vrací se hodnota posledního vyhodnoceného výrazu v těle funkce, tedy

```

> a = ahoj()

[1] "AHOJ,"
[1] "pane"

> a

[1] "pane"

```

srovnej s

```

> ahoj = function(jmeno = 'pane'){
+     return(c('AHOJ,', jmeno))
+     print('konec.')
+ }
> ahoj()

[1] "AHOJ," "pane"

```

V rámci definice funkce je možné použít speciální argument ..., který je posléze možné předat jiné funkci, např.

```

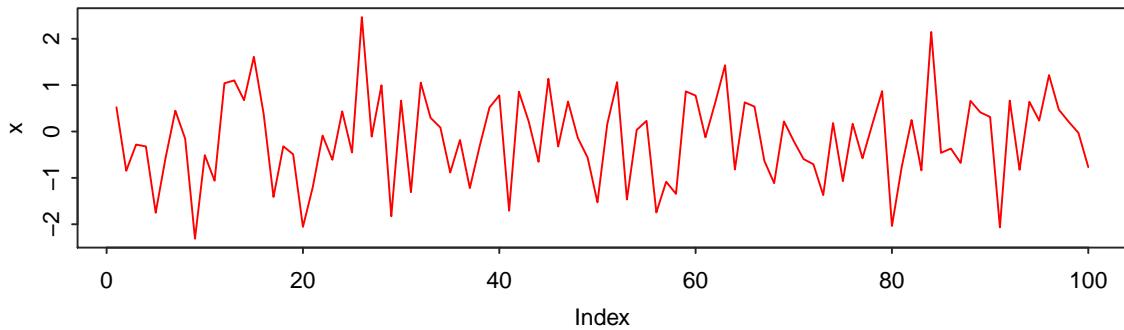
> plot_rnd = function(n, ...){
+     x = rnorm(n, 0, 1)
+     plot(x, type='l', ...)
+ }

```

Obr. 2.5: Demonstrace funkce argumentu ...

Funkce `plot_rnd` má argument `n`, který určuje, kolik náhodných čísel se bude generovat, argument `...` se předává funkci `plot`, tedy jakékoli argumenty jiné než `n` jsou použity ve funkci `plot`. Např.

```
> plot_rnd(100, col = 'red')
```



Podmínky

Při programování se často setkáváme se situací, kdy v závislosti na splnění nějaké podmínky chceme vykonat ten či jiný výraz - např. při řešení kvadratické rovnice počítáme kořeny dle hodnoty diskriminantu. V Rku využijeme pro tyto účely strukturu

```
> if (podmínka) výraz1 else výraz2
```

kdy pokud je splněna podmínka provede se výraz1 pokud ne, provede se výraz2, přičemž část od klíčového slova else je nepovinná. Výrazy mohou být značně rozsáhlé, např. mohou obsahovat další podmínky apod.

ÚKOL 2.25 Vypište, zdali je suma 100 náhodných čísel kladná nebo záporná.

```
> x = rnorm(100)
> sum(x)
[1] -18.38
> if (sum(x) > 0) ('kladne') else ('zaporene')
[1] "zaporene"
```

ÚKOL 2.26 Vypište, zdali jsou všechna x větší než -2?

```
> if (all(x > -2)) {znam = 'vse > -2'} else {znam = 'nejake < -2'}
> znam
[1] "nejake < -2"
```

ÚKOL 2.27 Vypište, zdali je nejake x větší než 3?

```
> if (any(x > 3)) {znam = 'vse < 3'} else {znam = 'nejake > 3'}
> znam
[1] "nejake > 3"
```

□

Podmínkou v příkazu if nemůže být vektor, respektive z vektoru je uvažován pouze první prvek. Pokud z nějakého důvodu potřebujeme podmínu aplikovat na jednotlivé prvky vektoru, je možné použít příkaz ifelse. Příkladem může být překódování číselného vektoru na vektor znakový (+, -) podle toho, je-li položka větší nebo menší než 0:

```
> ifelse(rnorm(10)>0, '+', '-')
[1] "+" "+" "-" "-" "-" "+" "-" "-" "+"
```

Cyklus for

Další velmi obvyklou situací je, že potřebujeme nějaký blok kódu opakovat, např. pro každý prvek z nějakého vektoru nebo *nkrát* apod. K tomu slouží struktura

```
> for (i in v) {
+     vyraz
+ }
```

tj. pro *i* nabývající hodnot z vektoru *v* prováděj vyraz. Proměnná *i* často figuruje v prováděném výrazu. Nejjednodušším příkladem může být vypsání čísel od 1 do 3

```
> for (i in 1:3){
+     print(i)
+ }
[1] 1
[1] 2
[1] 3
```

nebo výpis informací o souborech v aktuálním adresáři změněných během posledního dne

```
> soubory = dir(pattern='skripta')
> for (f in soubory){
+     if (file.info(f)$mtime>=(Sys.time() - 3600*24)) cat(f, file.info(f)$size, '\n')
+ }
skripta-concordance.tex 66
skripta.Rnw 11075
skripta.aux 1380
skripta.bbl 1038
skripta.log 133798
skripta.pdf 1765321
skripta.synctex.gz 713786
skripta.tex 12906
skripta.toc 4248
```

2.8 Základní grafika

Na začátku kapitoly 2.4 jsme vykreslili graf pomocí funkce *plot*, jejímž jediným parametrem byl vektor srážek (*rain*) – v tom případě hodnoty na ose x představují pořadí (index) v rámci vektoru, hodnoty na ose y pak čísla ve vektoru obsažená. Tento typ grafu se nazývá *index plot* - indexový graf. Příkazy

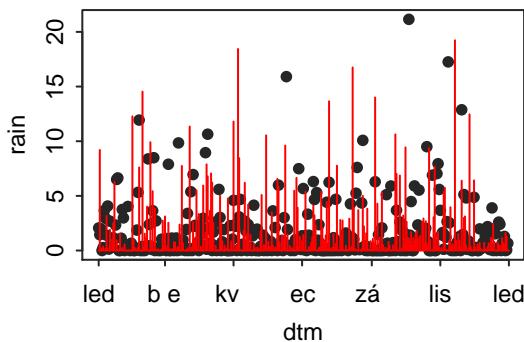
```
> plot(rain)
> plot(y=rain)
> plot(x=1:length(rain), y=rain)
> plot(1:length(rain), rain)
```

jsou ekvivalentní. V RStudiu se graf objeví v okně **Plots**, kde je k dispozici i historie kreslených obrázků. Pokud chceme mít na ose x přímo datum vztahující se k časové řadě, je nutné předat funkci

jako parametry jak datum (parametr x), tak srážky (parametr y).

Funkce `plot` ve výchozím nastavení vždy smaže stávající graf a nakreslí nový. Pokud chceme v jednom obrázku znázornit více časových řad, použijeme funkci `lines`. Řada se zobrazí pomocí čáry, stejně jako při použití parametru `type = "l"` ve funkci `plot`. (K zobrazení bodů slouží analogická funkce `points`.)

```
> rain2 = 6 * rgamma(365, shape=.3)
> rain2[rain2<.05] = 0 # srážky pod 0.05 mm prohlásíme za nulové
> plot(dtm, rain, pch=20)
> lines(dtm, rain2, col='red', type='h')
```



Obr. 2.6: Ukázka tvorby grafu.

Parametr `col` udává barvu; platné názvy barev vypíše funkce `colors`. Barvy lze zadávat také v hexadecimálním tvaru nebo jako kombinace barevných složek. Kromě barvy mohou mít grafické funkce množství dalších parametrů, které určují vzhled grafu. Nejužitečnější z nich udává následující tabulka:

Tab. 2.4: Nejpoužívanější grafické parametry.

parametr	význam
<code>type</code>	typ zobrazení (body "p", čáry "l", body i čáry "b" atd.)
<code>col</code>	barva
<code>lty</code>	typ čáry (plná, čárkovaná apod.) – číselné hodnoty od 1
<code>lwd</code>	šířka čáry
<code>pch</code>	vzhled bodu (kolečko, čtverec, hvězdička apod.) – číselné hodnoty od 1
<code>xlim</code>	rozsah osy x, dvouprvkový vektor se začátkem a koncem rozsahu
<code>ylim</code>	rozsah osy y
<code>xlab</code>	popisek osy x
<code>ylab</code>	popisek osy y
<code>main</code>	nadpis grafu

Globálně (pro všechny následující obrázky) se dají parametry změnit pomocí funkce `par`. Legenda se vytváří zvlášť funkcí `legend`:

```
> legend("topright", c("rain1", "rain2") , col = c("black", "red"), pch = c(20, NA), lty = c(NA, 1),
+         title = "rain")
```

Další typy grafů jsou uvedeny průběžně v rámci následujících kapitol. Mimo typů grafů obsažených v základní distribuci Rka existuje velké množství balíků zprostředkovávajících funkce pro tvorbu speciálních typů grafů, map (`raster`, `rasterVis`, `RGoogleMaps`), balíky pomáhající při efektivní

vizuální analýze dat (`lattice`, `grid`, `ggplot2`), případně balíky umožňující tvorbu interaktivních aplikací (`shiny`, `ggvis`).

Export grafu

Obrázky se v Rku vykreslují standardně na obrazovku. Pro další využití je můžeme v RStudiou uložit jako soubor příkazy z okna Plots: **Export – Save Plot as Image** nebo **Export – Save Plot as PDF**. Pohodlnější je však ukládat obrázky přímo v Rku za pomoci příslušných funkcí. Například graf v souboru s rastrovým obrázkem typu PNG vytvoříme jako

```
> png("graf1.png", width = 800, height = 600)
> # ...
> # kód vytvářející graf
> # ...
> dev.off()
```

Funkce `dev.off` uzavírá výstup na určité výstupní zařízení (*device*) – v tomto případě výstup do souboru, který se otevřel vykonáním funkce `png`. (Díky systému výstupních zařízení je možné například průběžně přidávat prvky do obrázků v různých souborech.) Vlastnosti exportovaného obrázku lze dále upravit parametry `res` (rozlišení v DPI - *dots per inch*) a `pointsizes` (velikost tiskového bodu). V návodě k funkci `png` jsou uvedeny i funkce pro jiné formáty rastrových obrázků (JPG, TIFF apod.).

Rko podporuje také vektorové formáty, například do souboru typu PDF se graf uloží jako

```
> pdf("graf1.pdf", width = 8, height = 6)
> # ...
> # kód vytvářející graf
> # ...
> dev.off()
```

kde `width` a `height` je v palcích.

2.9 Vybraná pokročilá téma

2.9.1 `data.table`

`Data.table` je struktura podobná `data.frame`, která umožnuje provádět pohodlně a efektivně některé operace, zejména při práci s velkým objemem dat. Některé rozdíly v syntaxi mezi `data.frame`m a `data.table`m udává následující tabulka. Nejdřív ale vytvořme `data.table` obsahující fiktivní srážky a datum měření a pro porovnání vytvořme i identický `data.frame`:

```
> require(data.table)
> dtm = seq(as.Date('2013-01-01'), as.Date('2015-12-31'), by = 'day')
> rain = 6 * rgamma(365, shape=.3)
> rain[rain<.05] = 0 # nastavíme malé srážky = 0
> dta = data.table(DTM = dtm, R = rain)
> dtf = data.frame(dta)
```

Tab. 2.5: Porovnání syntaxe data.framu a data.tablu.

	data.frame	data.table
výběr veličiny R	dtf[, "R"] dtf\$R dtf[["R"]]	dta[, R] dta\$R dta[["R"]]
výběr veličin DTM a R	dtf[, c("DTM", "R")]	dta[, list(DTM, R)]
výpočet R + log(R)	dtf\$R + log(dtf\$R)	dta[, R + log(R)]
výběr pouze řádků, pro které platí, že R>10	dtf[dtf\$R>10,]	dta[R>10,]
zařazení nové proměnné R2 = R - exp(R)	dtf\$R2 = dtf\$R - exp(dtf\$R)	dta[, R2:= R - exp(R)]

Data.table navíc pomocí argumentu by umožnuje pohodlné vyhodnocení funkcí (např. extrakci maxim) pro různé skupiny jevů, např. pro jednotlivé roky. Data.table disponuje funkcemi pro extrakci roků (year()), měsíců (month()), čtvrtletí (quarter()), týdnů (week()), dnů v týdnu (wday()) apod. z objektů třídy datum (viz ?year). Například průměrnou srážku pro jednotlivé dny v týdnu spočítáme pomocí

```
> dta[, mean(R), by = wday(DTM)]
   wday      V1
1:   3 1.524
2:   4 1.726
3:   5 1.838
4:   6 1.878
5:   7 1.711
6:   1 1.727
7:   2 1.666
```

Skupiny je možno kombinovat, např. měsíční průměrné průtoky pro jednotlivé roky získáme pomocí

```
> dta[, mean(R), by = list(month(DTM), year(DTM))]
```

ÚKOL 2.28 Jak se liší výsledné objekty?

- dta[, mean(R), by = month(DTM)]
- dta[, list(R = mean(R)), by = month(DTM)]
- dta[, mean(R), by = list(MESIC = month(DTM))]

□

2.9.2 dplyr

bude součástí další verze skript

2.9.3 DateTime třídy

bude součástí další verze skript

2.9.4 ggplot2

bude součástí další verze skript

2.9.5 GIS v R

bude součástí další verze skript

3 Náhodné veličiny

Náhodná veličina je taková veličina, která mění své hodnoty v závislosti na náhodě. Někdy je náhodné kolísání výsledků dánno existencí náhodných chyb měření. Měříme-li například průtok v toku, je vždy měření zatíženo nepřesností měření. V jiných případech je náhodnost přímo obsažená v daných jevech - například naměřená hodnota průtoku v toku bude pokaždě jiná v závislosti na stavu povodí.

Rozlišujeme dva typy náhodných veličin - diskrétní a spojité. Diskrétní náhodná veličina může nabývat konečně nebo spočetně hodnot, v aplikacích se dá často vyjádřit jako počet, např. počet let za století, kdy průtok přesáhl určitou mez. Spojitá náhodná veličina může nabývat všech hodnot z určitého intervalu, např. okamžitý průtok v řece (Jarušková, 2011).

3.1 Distribuční a kvantilová funkce, hustota pravděpodobnosti

Distribuční funkce

Valnou většinu veličin, se kterými se setkáváme v hydrologii můžeme považovat za (spojité) náhodné veličiny. Jakkoliv veličina X je náhodnou veličinou, pokud pro nějakou hodnotu x , kterou může veličina nabývat, existuje pravděpodobnost P , že daná veličina je menší nebo rovna hodnotě x , $P(X \leq x)$. Dále platí, že

$$0 \leq P(X \leq x) \leq 1 \quad (3.1)$$

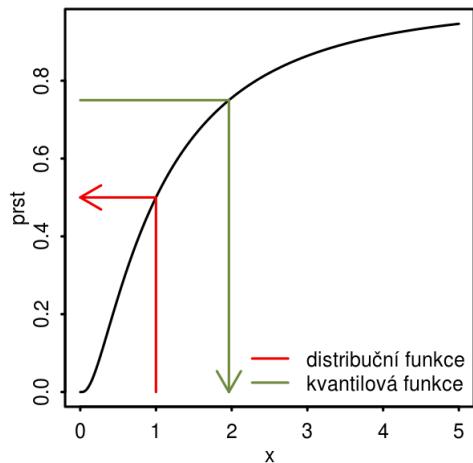
Pravděpodobnost $P(X \leq x)$ se nazývá distribuční funkce $F(x)$. Náhodná veličina je charakterizována množinou jevů, tj. množinou hodnot, které může nabývat, a pravděpodobností jednotlivých jevů, respektive distribuční funkcí (Yevjevich et al., 1972). Například pro náhodnou veličinu X - průtok Vltavy v Podbabě jsou množinou jevů nezáporná čísla a zároveň platí, že existuje pravděpodobnost $0 \leq P(X \leq x) \leq 1$, že průtok nabývá hodnot stejných nebo nižších než x , pro libovolnou hodnotu x z množiny jevů (např. průtok $100 \text{ m}^3/\text{s}$).

Pro distribuční funkci $F(x)$ platí, že

- pro všechny hodnoty $x \in (-\infty, \infty)$ je $0 \leq F(x) \leq 1$
- $F(x)$ je neklesající
- pro $x_1 < x_2$ je $P(x_1 < X \leq x_2) = F(x_2) - F(x_1)$

Kvantilová funkce

Inverzní funkcí k distribuční funkci je funkce kvantilová (viz obr. 3.1). Tato funkce přiřazuje hodnoty na základě pravděpodobnosti - tj. p -procentní kvantil je taková hodnota q_p , pro kterou platí, že $P(X \leq q_p) = p/100$. Např. 90% kvantil náhodné veličiny je taková hodnota, pro kterou platí, že pravděpodobnost, že veličina nabývá stejných nebo nižších hodnot, je 90 %.



Obr. 3.1: Distribuční a kvantilová funkce

Hustota rozdělení pravděpodobnosti

Další důležitou charakteristikou náhodné veličiny je hustota rozdělení pravděpodobnosti. Hustota pravděpodobnosti spojité náhodné veličiny X je funkce f , pro kterou platí, že

- 1 $f(x) \geq 0$ pro všechna $x \in (-\infty, \infty)$, tj. pravděpodobnost pro libovolné x je buď 0 nebo kladné číslo, tedy funkce je nezáporná
- 2 $\int_{-\infty}^{\infty} f(x) = 1$ - integrál hustoty pravděpodobnosti je 1
- 3 $f(x) = F'(x)$ - hustota je derivací distribuční funkce (a distribuční funkce je integrálem hustoty)
- 4 $P(x \in M) = \int_{x \in M} f(x)$ pro libovolnou množinu reálných čísel M - pravděpodobnost, že x leží v množině M , je rovna integrálu pravděpodobnosti přes tuto množinu
- 5 $P(x = c) = 0$ pro každé $c \in (-\infty, \infty)$ - pravděpodobnost, že veličina nabývá konkrétní hodnoty, je rovna 0.

Praktická znalost toho, jakým rozdělením se určitá náhodná veličina řídí, je obvykle dána dlouhodobou zkušeností. Někdy se typ rozdělení dá odvodit z teoretických úvah (Jarušková, 2011). Z tohoto pohledu můžeme rozdělení nějaké náhodné veličiny považovat za teoretické (založeno na axiomech a odvozeních využívajících teorii pravděpodobnosti), semi-teoretická (máme-li např. indície, že daná veličina by měla mít rozdělení z nějaké rodiny rozdělení pravděpodobnosti) a empirická (odhadnutá čistě na základě výběru, např. měření), srov. Yevjevich et al. (1972).

Ve statistice a jejích aplikacích v oblasti vodního hospodářství se běžně využívá řada rozdělení pravděpodobnosti, např. normální, logaritmicko-normální, gama, beta, exponenciální, Gumbelovo, χ^2 , Studentovo-t, F rozdělení ad. Skutečnost, že náhodná veličina X má nějaké specifické rozdělení, se často vyjadřuje zápisem $X \sim D(\theta)$, kde D označuje rozdělení a θ je vektor parametrů. Například zápis $X \sim N(0, 1)$ znamená, že náhodná veličina X má normální rozdělení se střední hodnotou 0 a rozptylem 1 (viz dále).

3.2 Základní charakteristiky rozdělení pravděpodobnosti

Základní charakteristikou náhodné veličiny je střední hodnota. Pro diskrétní náhodnou veličinu (tj. takovou, která může nabývat pouze konečného počtu hodnot) $X = (x_1, x_2, \dots, x_N)$ je střední hodnota definována jako

$$E(X) = x_1 p_1 + x_2 p_2 + \dots + x_N p_N = \sum_i x_i p_i, \quad (3.2)$$

kde p_i je pravděpodobnost $P(X = x_i)$. V podstatě jde tedy o vážený průměr možných hodnot.

Pro spojitou veličinu je střední hodnota definována obdobně pomocí integrálu

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx. \quad (3.3)$$

Základní vlastnosti rozdělení pravděpodobnosti je možno vyjádřit několika veličinami, které souhrnně označujeme pojmem momentové charakteristiky řádu k . Momenty mohou být

obecné

$$\mu_k = E(X^k) = \int_{-\infty}^{\infty} x^k f(x) dx \quad (3.4)$$

centrální

$$\mu'_k = E[(X - E(X))^k] = \int_{-\infty}^{\infty} (x - \mu_1)^k f(x) dx \quad (3.5)$$

standardizované

$$\mu''_k = \frac{\mu'_k}{(\mu'_2)^{k/2}} \quad (3.6)$$

Typicky se pro charakterizaci náhodné veličiny používá kombinace různých typů momentů, viz Tabulka 3.1. Vyjmenovaným momentům z tabulky se věnujeme dále.

Tab. 3.1: Momenty a charakteristiky náhodné veličiny.

Řád momentu	Obecný moment	Centrální moment	Standardizovaný moment
1	střední hodnota	0	0
2	-	rozptyl	1
3	-	-	šikmost
4	-	-	špičatost

Poloha

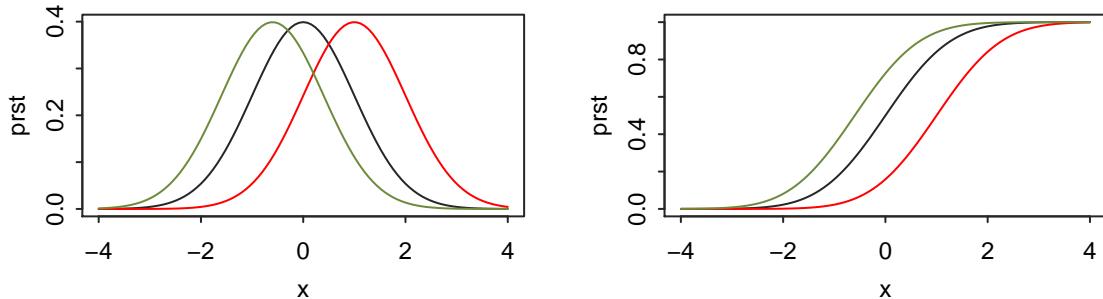
Polohu, těžiště rozdělení pravděpodobnosti náhodné veličin X lze vyjádřit pomocí střední hodnoty (prvního obecného momentu)

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx \quad (3.7)$$

Dále platí, že

- $E(aX + b) = aE(X) + b$, pro $a, b \in (-\infty, \infty)$

- střední hodnota součtu náhodných veličin je rovna součtu středních hodnot těchto veličin, tj.
 $E(\sum_i X_i) = \sum_i E(X_i)$, jsou-li veličiny nezávislé
- střední hodnota součinu náhodných veličin je rovna součinu středních hodnot těchto veličin, tj.
 $E(\prod_i X_i) = \prod_i E(X_i)$, jsou-li veličiny nezávislé



Obr. 3.2: Rozdělení lišící se polohou - hustota (vlevo) a distribuční funkce (vpravo).

Variabilita

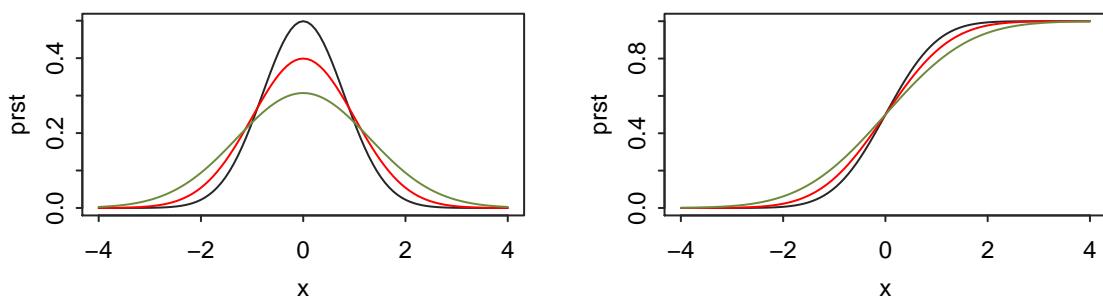
Variabilitu náhodné veličiny X je možno vyjádřit pomocí rozptylu (tj. druhého centrálního momentu)

$$V(X) = E([X - E(X)]^2) \quad (3.8)$$

Rozptyl je tedy střední hodnota odchylek od střední hodnoty veličiny X .

Dále platí, že

- $V(X) \geq 0$, tj. rozptyl je nezáporný
- rozptyl konstanty c je $V(c) = 0$ pro všechna $c \in (-\infty, \infty)$
- $V(X) = E(X^2) - [E(X)]^2$
- $V(aX + b) = a^2 V(X)$
- rozptyl sumy náhodných veličin je roven sumě rozptylů těchto veličin, tj. $V(\sum_i X_i) = \sum_i V(X_i)$, jsou-li veličiny X_i nezávislé



Obr. 3.3: Rozdělení lišící se variabilitou - hustota (vlevo) a distribuční funkce (vpravo).

Tvar

Tvar lze charakterizovat pomocí šikmosti (třetí standardizovaný moment)

$$\mu_3 = \frac{E([X - E(X)]^3)}{[V(X)]^{3/2}} \quad (3.9)$$

- $\mu_3 = 0$ pro symetrické rozdělení
- $\mu_3 < 0$ pro zprava zešikmené rozdělení
- $\mu_3 > 0$ pro zleva zešikmené rozdělení

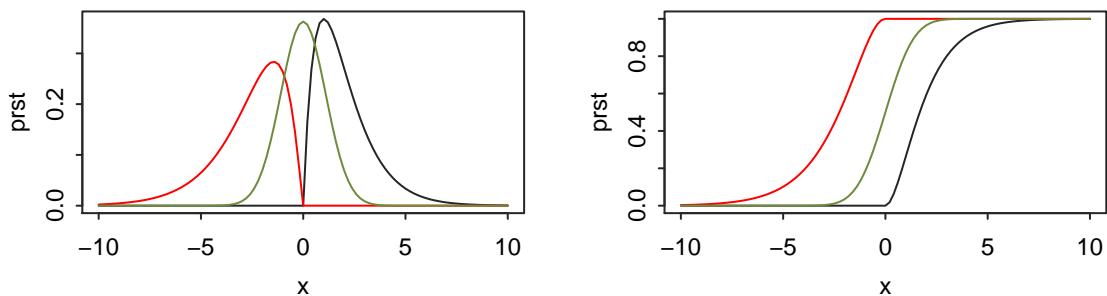
a špičatosti (čtvrtý standardizovaný moment)

$$\mu_4 = \frac{E([X - E(X)]^4)}{[V(X)]^{4/2}} \quad (3.10)$$

Koeficient špičatosti bývá také někdy zapisován jako

$$\mu_4 = \frac{E([X - E(X)]^4)}{[V(X)]^{4/2}} - 3 \quad (3.11)$$

kde -3 je interpretováno jako korekce za účelem dosažení nulového koeficientu špičatosti pro normální rozdělení. Tato verze koeficientu se někdy nazývá *excess kurtosis* (Yevjevich et al., 1972).

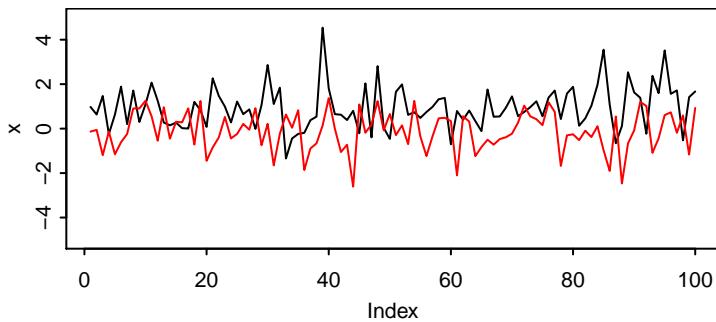


Obr. 3.4: Rozdělení lišící se tvarem - hustota (vlevo) a distribuční funkce (vpravo).

Kódy v následujících příkladech jsou vysvětleny v sekci 3.3.

ÚKOL 3.1 Čím se liší veličiny na obrázku?

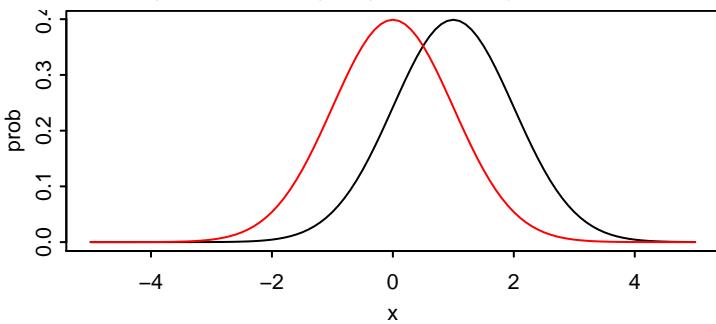
```
> plot(rnorm(100, mean = 1), ylab = 'x', type='l', ylim = c(-5, 5))
> lines(rnorm(100), col='red')
```



Obr. 3.5: Obrázek k příkladu 3.1.

ÚKOL 3.2 Čím se liší veličiny na obrázku?

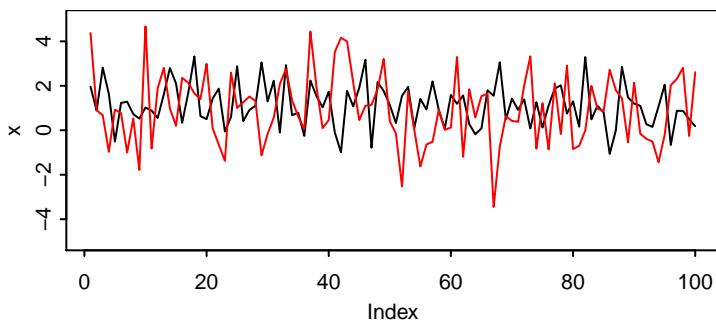
```
> curve(dnorm(x, mean = 1), xlim = c(-5, 5), ylab = 'prob')
> curve(dnorm(x), xlim = c(-5, 5), col='red', add=TRUE)
```



Obr. 3.6: Obrázek k příkladu 3.2.

ÚKOL 3.3 Čím se liší veličiny na obrázku?

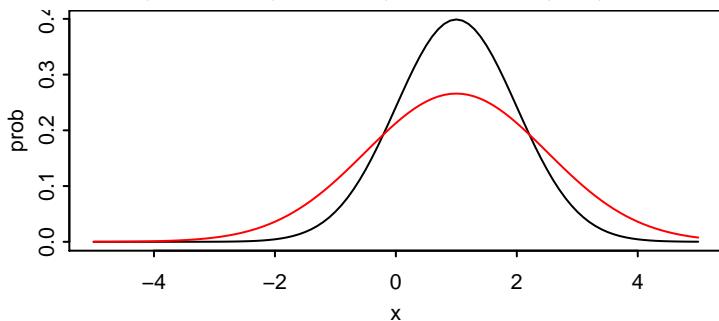
```
> plot(rnorm(100, mean = 1), ylim = c(-5, 5), ylab = 'x', type='l')
> lines(rnorm(100, mean = 1, sd=1.5), col='red')
```



Obr. 3.7: Obrázek k příkladu 3.3.

ÚKOL 3.4 Čím se liší veličiny na obrázku?

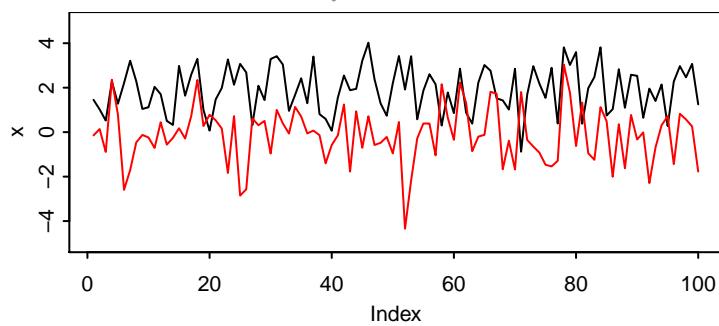
```
> curve(dnorm(x, mean = 1), xlim = c(-5, 5), ylab = 'prob')
> curve(dnorm(x, mean = 1, sd=1.5), xlim = c(-5, 5), col='red', add=TRUE)
```



Obr. 3.8: Obrázek k příkladu 3.4.

ÚKOL 3.5 Čím se liší veličiny na obrázku?

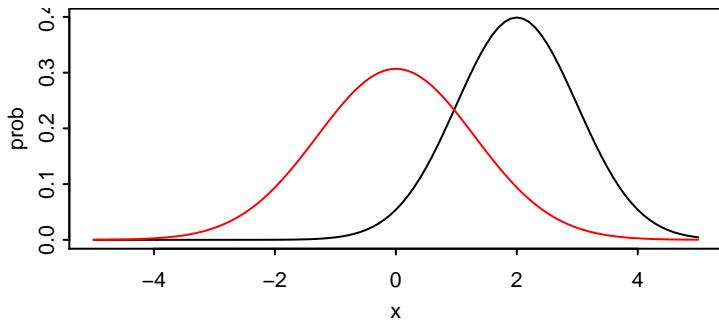
```
> plot(rnorm(100, mean = 2), ylim = c(-5, 5), ylab = 'x', type='l')
> lines(rnorm(100, sd = 1.3), ylim = c(-5, 5), col='red')
```



Obr. 3.9: Obrázek k příkladu 3.5.

ÚKOL 3.6 Čím se liší veličiny na obrázku?

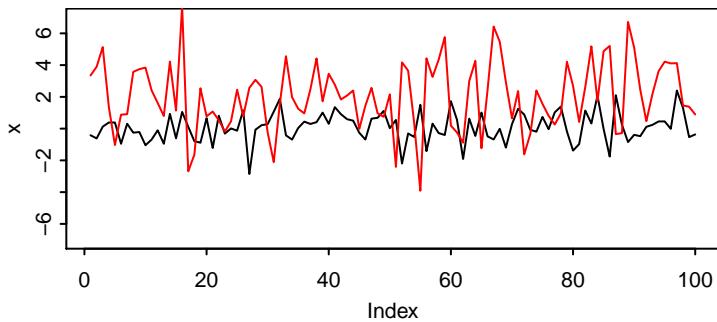
```
> curve(dnorm(x, mean = 2), xlim = c(-5, 5), ylab = 'prob')
> curve(dnorm(x, sd = 1.3), xlim = c(-5, 5), col='red', add=TRUE)
```



Obr. 3.10: Obrázek k příkladu 3.6.

ÚKOL 3.7 Čím se liší veličiny na obrázku?

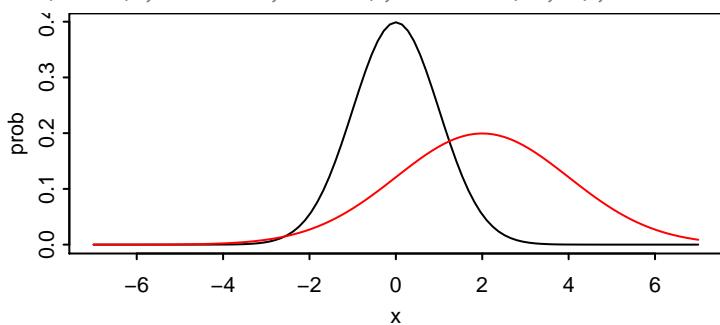
```
> plot(rnorm(100), ylim = c(-7, 7), ylab = 'x', type='l')
> lines(rnorm(100, mean = 2, sd = 2), col='red')
```



Obr. 3.11: Obrázek k příkladu 3.7.

ÚKOL 3.8 Čím se liší veličiny na obrázku?

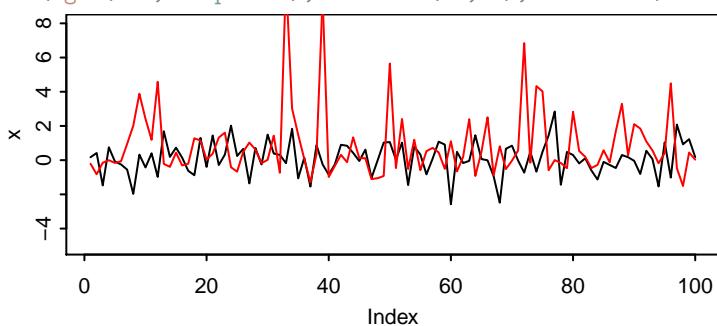
```
> curve(dnorm(x), xlim = c(-7, 7), ylab = 'prob')
> curve(dnorm(x, mean = 2, sd = 2), xlim = c(-7, 7), col='red', add=TRUE)
```



Obr. 3.12: Obrázek k příkladu 3.8.

ÚKOL 3.9 Čím se liší veličiny na obrázku?

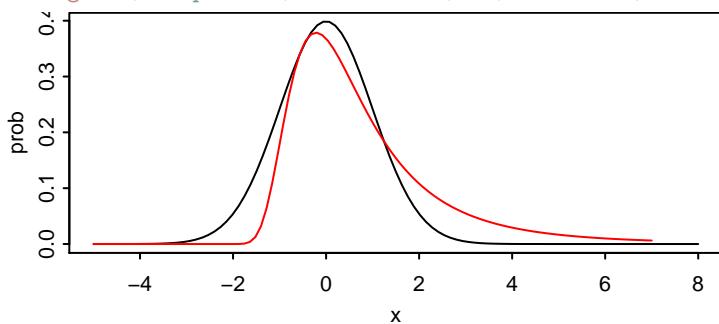
```
> require(evd)
> plot(rnorm(100), ylim = c(-5, 8), ylab = 'x', type='l')
> lines(rgev(100, shape=.25), xlim = c(-5, 7), col='red')
```



Obr. 3.13: Obrázek k příkladu 3.9.

ÚKOL 3.10 Čím se liší veličiny na obrázku?

```
> require(evd)
> curve(dnorm(x), xlim = c(-5, 8), ylab = 'prob')
> curve(dgev(x, shape=.25), xlim = c(-5, 7), col='red', add=TRUE)
```



Obr. 3.14: Obrázek k příkladu 3.10.

□

3.3 Práce s rozděleními v R

Rko zná celou řadu rozdělení náhodných veličin již ve výchozí instalaci (jsou součástí balíku stats, pro jejich přehled viz ?Distributions). Řada dalších rozdělení je obsažena v doplňkových balíčcích.

Funkce pro práci s rozděleními mají formu

dxxx(x, ...)	hustota pravděpodobnosti
pxxx(q, ...)	distribuční funkce
qxxx(p, ...)	kvantilová funkce
rxxx(n, ...)	generátor náhodných čísel z daného rozdělení

kde xxx je akronym rozdělení pravděpodobnosti např. norm pro normální rozdělení exp pro expo-nenciální atd. (viz ?Distributions). Argumenty funkcí jsou kvantily veličiny (x, q), pravděpodobnosti (p) nebo počet generovaných čísel (n). Za argument ... se mohou doplnit parametry jednotlivých rozdělení, např. v případě normálního rozdělení střední hodnota (mean) a směrodatná odchylka (sd). Pokud parametry rozdělení nejsou zadány, doplní Rko výchozí hodnoty (existují-li), které zpravidla odpovídají standardní formě daného rozdělení (např. standardní normální rozdělení má střední hodnotu 0 a rozptyl 1). Požadované a výchozí hodnoty parametrů lze zjistit buď v návodě nebo pomocí funkce args:

```
> args(qnorm)
function (p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
NULL
> args(qbeta)
function (p, shape1, shape2, ncp = 0, lower.tail = TRUE, log.p = FALSE)
NULL
```

Pokud má parametr výchozí hodnotu, je to uvedeno (např. mean = 0), pokud nemá, figuruje v definici funkce pouze jméno argumentu (např. shape1). V tom případě Rko očekává, že hodnotu zadáme.

Tedy např.

```

> # 25% kvantil standardního normálního rozdělení
> qnorm(.25)
[1] -0.6745

> # 25% a 75% kvantil normálního rozdělení se střední hodnotou 1 a směrodatnou odchylkou 1.5
> qnorm(c(.25, .75), mean = 1, sd = 1.5)
[1] -0.01173  2.01173

> # 25% kvantil beta rozdělení - chyba - nezadali jsme argumenty, které Rku potřebuje
> qbeta(.25)

Error: argument "shape1" is missing, with no default

> # správně
> qbeta(.25, shape1=1, shape2=0.5)
[1] 0.4375

```

ÚKOL 3.11 Zjistěte 10, 20, 30, 40, 50, 60, 70, 80, 90% kvantil normálního rozdělení se střední hodnotou -1 a směrodatnou odchylkou 3.

```

> qnorm(seq(.1, .9, by = .1), mean = -1, sd=3)
[1] -4.8447 -3.5249 -2.5732 -1.7600 -1.0000 -0.2400  0.5732  1.5249  2.8447

```

ÚKOL 3.12 Zjistěte, jaká je pravděpodobnost, že veličina $X \sim N(0, 1)$ je nižší než 1, vyšší než 4. Využijte definici distribuční funkce a příkaz pnorm.

```

> pnorm(1)
[1] 0.8413

> 1-pnorm(4)
[1] 3.167e-05

```

ÚKOL 3.13 Zjistěte, jaká je pravděpodobnost, že veličina $X \sim N(10, 5)$ leží v intervalu (15, 20).

```

> pnorm(20, mean = 10, sd = 5) - pnorm(15, mean = 10, sd = 5)
[1] 0.1359

```

□

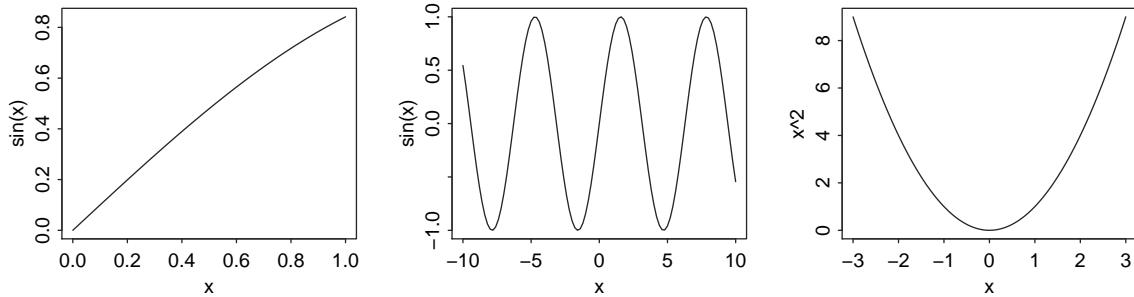
Vizualizace rozdělení pravděpodobnosti

Jakékoli funkce lze v Rku vykreslit pomocí funkce `curve`. Rozsah hodnot, pro které je funkce vyčíslena, je dán argumentem `xlim`. Pomocí `curve` lze vykreslit standardní funkce (nebo funkce definované uživatelem), obecně lze vykreslit jakoukoliv funkci proměnné `x`:

```

> # vykreslím funkci sinus
> curve(sin(x))
> curve(sin(x), xlim = c(-10, 10))
>
> # vykreslím křivku x^2
> curve(x^2, xlim = c(-3, 3))

```



Obr. 3.15: Ukázka práce s funkcí `curve`.

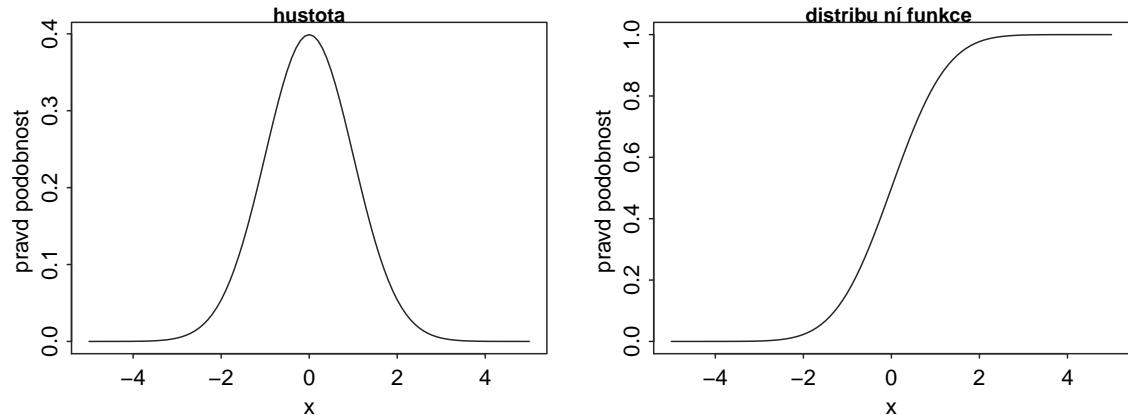
Výchozím předpokladem v Rku je, že funkce se vztahuje k proměnné `x`, nicméně není-li tomu tak, je možno specifikovat název proměnné pomocí argumentu `xname`. Například závislost velikosti 95% kvantilu lognormálního rozdělení na směrodatné odchylce (v rozmezí 0 a 3) lze vykreslit pomocí `curve(qlnorm(p=.95, sd = std), xname='std', xlim = c(0, 3))`.

Hustotu pravděpodobnosti a distribuční funkci normálního rozdělení můžeme tedy jednoduše vykreslit pomocí:

```

> curve(dnorm(x), xlim = c(-5, 5), main = 'hustota', ylab = 'pravděpodobnost', cex.main = .9)
> curve(pnorm(x), xlim = c(-5, 5), main = 'distribuční funkce', ylab = 'pravděpodobnost', cex.main = .9)

```



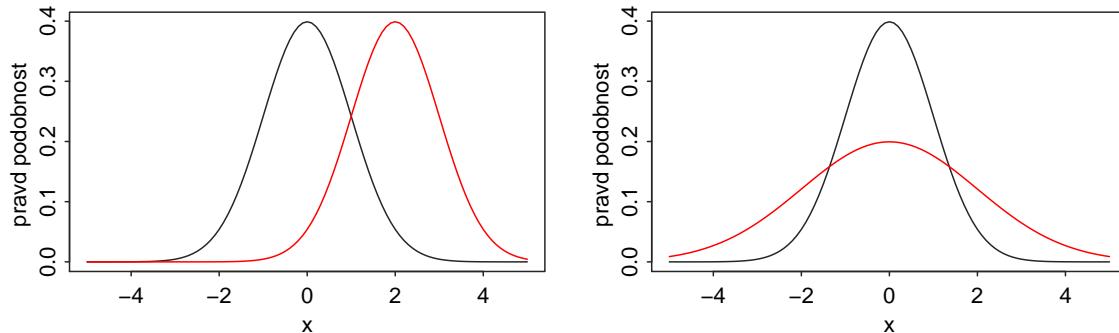
Obr. 3.16: Vykreslení hustoty a distribuční funkce normálního rozdělení.

Pokud místo kreslení dalšího grafu chceme přidat křivku do grafu stávajícího, použijeme argument `add = TRUE`. Chceme-li měnit parametry vykreslované funkce (např. změnit střední hodnotu rozdělení), jednoduše zahrneme tyto parametry do vykreslované funkce:

```

> # graf vlevo
> curve(dnorm(x), xlim = c(-5, 5), ylab = 'pravděpodobnost')
> curve(dnorm(x, mean=2), col='red', add=TRUE)
>
> # graf upravo
> curve(dnorm(x), xlim = c(-5, 5), ylab = 'pravděpodobnost')
> curve(dnorm(x, sd=2), col='red', add=TRUE)

```



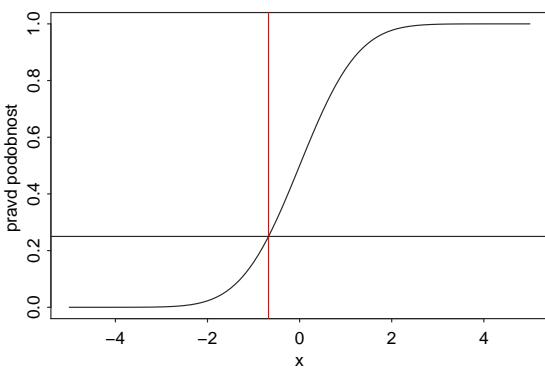
Obr. 3.17: Změna parametrů vykreslovaných rozdělení.

Do grafů je možné pomocí příkazu abline přidávat horizontální (abline(h = X)) a vertikální (abline(v = Y)) čáry, např. pro zvýraznění určitých bodů:

```

> # distribuční funkce standardního normálního rozdělení
> curve(pnorm(x), xlim = c(-5, 5), ylab = 'pravděpodobnost')
> # zvýrazni 25% kvantil
> abline(v = qnorm(.25), col='red')
> # zvýrazni pravděpodobnost (tedy 0.25)
> abline(h = .25)

```



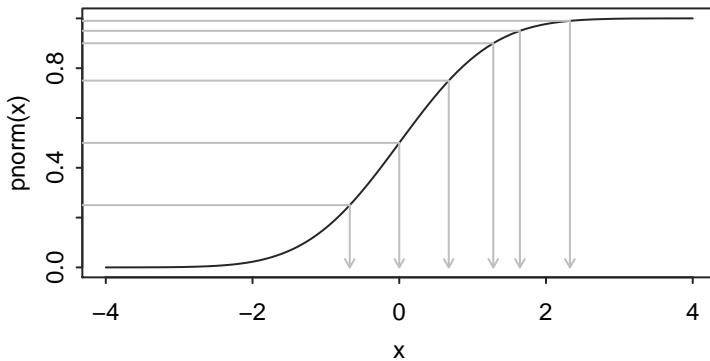
Obr. 3.18: Přidávání čar do grafu.

ÚKOL 3.14 Vykreslete distribuční funkci standardního normálního rozdělení a vyznačte v něm jednotlivé quartily a 90, 95 a 99% kvantil. Jaké jsou jejich hodnoty?

```

> curve(pnorm(x), xlim = c(-4, 4))
> p = c(.25,.5,.75, .9, .95, .99)
> q = qnorm(p)
> q
[1] -0.6745  0.0000  0.6745  1.2816  1.6449  2.3263
> segments(-10, p, q, p, col='gray')
> arrows(q, p, q, 0, length=0.05, col='gray')

```



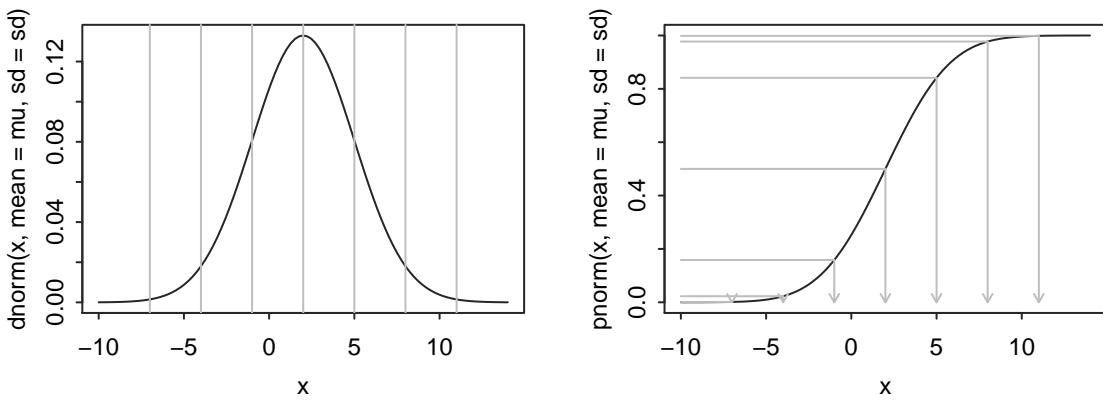
Obr. 3.19: Řešení.

ÚKOL 3.15 Vykreslete hustotu a distribuční funkci veličiny X , která má normální rozdělení se střední hodnotou $\mu = 2$ a směrodatnou odchylkou $\sigma = 3$. Vyznačte střední hodnotu a dále vyznačte body $\mu \pm (\sigma, 2\sigma, 3\sigma)$. Zjistěte, jaká je pravděpodobnost, že $X < \mu - \sigma$, $X < \mu - 2\sigma$ a $X < \mu - 3\sigma$, respektive, že $X > \mu + \sigma$, $X > \mu + 2\sigma$ a $X > \mu + 3\sigma$?

```

> mu = 2
> sd = 3
> q = mu + (-3:3) * sd
> p = pnorm(q, mean=mu, sd=sd)
> round(100*c(p[1:4], 1-p[5:7]),2)
[1] 0.13 2.28 15.87 50.00 15.87 2.28 0.13
> curve(dnorm(x, mean = mu, sd=sd), xlim = c(mu - 4*sd, mu+4*sd))
> abline(v = q, col='gray')
>
> curve(pnorm(x, mean=mu, sd=sd), xlim = c(mu - 4*sd, mu+4*sd))
> segments(-10, p, q, p, col='gray')
> arrows(q, p, q, 0, length=0.05, col='gray')

```



Obr. 3.20: Řešení.

Uvědomte si, že výsledek je obecný - tj. (v případě normálního rozdělení) platí pro libovolné μ a σ . Dá se tedy říct, že v případě normálního rozdělení je pravděpodobnost, že odchylka od průměru je vyšší než 1 směrodatná odchylka, je 16 %, vyšší než 2 směrodatné odchylky 2 % a vyšší než 3 směrodatné odchylky cca 0.1 %. Orientačně si výsledek zapamatujte. Tato skutečnost může sloužit pro jednoduché vyhodnocení pravděpodobnosti extrémních jevů.

□

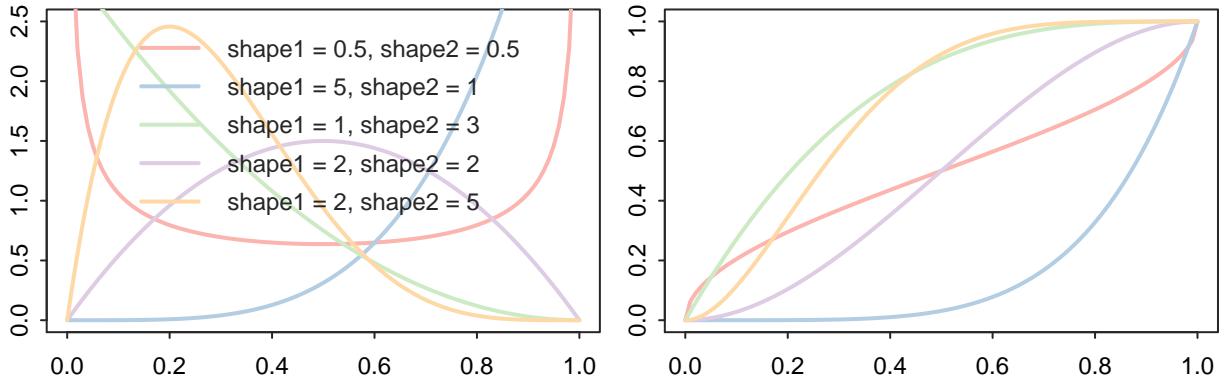
3.4 Vybraná rozdělení v R

Beta rozdělení

$$F(x; \alpha, \beta) = \frac{B(x; \alpha, \beta)}{B(\alpha, \beta)} \quad (3.12)$$

$$f(x; \alpha, \beta) \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (3.13)$$

B je beta funkce, v R $\alpha = \text{shape1}$, $\beta = \text{shape2}$ a funkce jsou ve formátu `xbeta`.



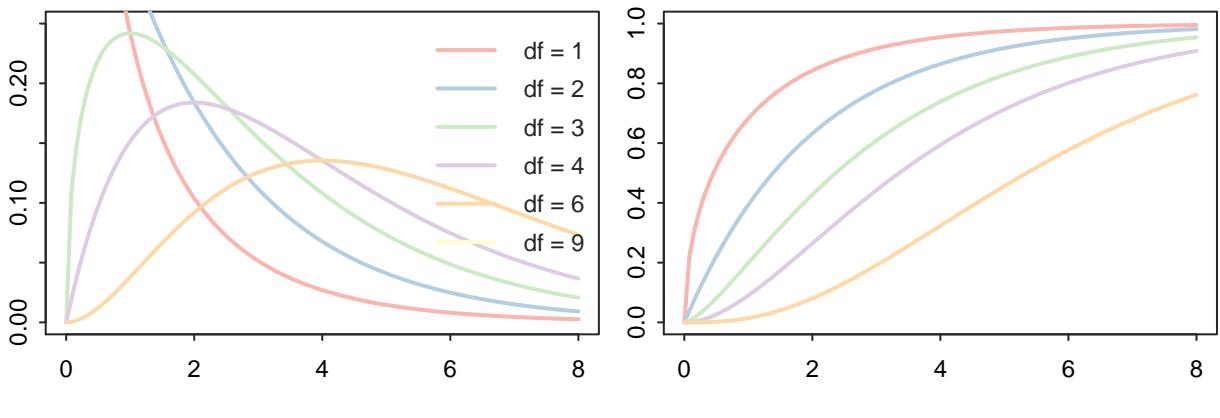
Obr. 3.21: Beta rozdělení.

χ^2 rozdělení

$$F(x) = \frac{1}{\Gamma(\frac{k}{2})} \gamma\left(\frac{df}{2}, \frac{x}{2}\right) \quad (3.14)$$

$$f(x) = \frac{1}{2^{\frac{df}{2}} \Gamma\left(\frac{df}{2}\right)} x^{\frac{df}{2}-1} \exp^{-\frac{x}{2}} \quad (3.15)$$

kde Γ je gama funkce. V R ve formátu `xchisq`.

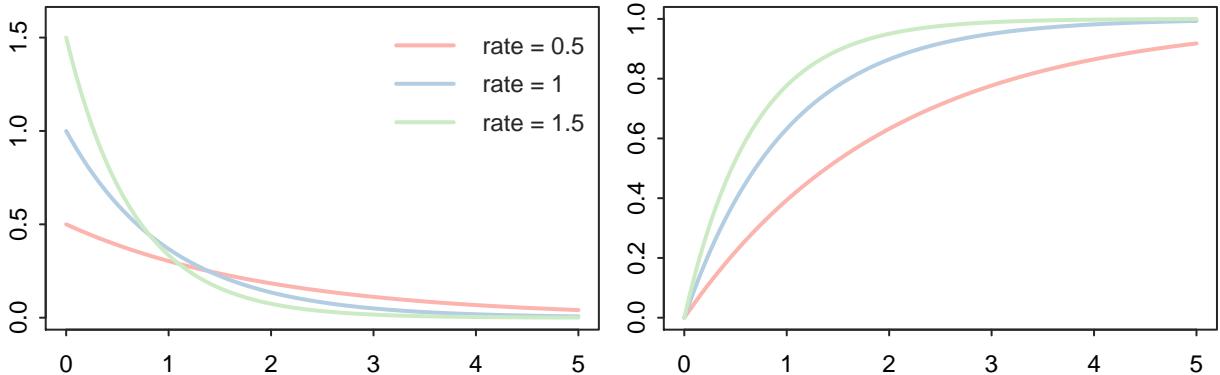


Obr. 3.22: χ^2 rozdělení.

Exponenciální rozdělení

$$F(x; \lambda) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases} \quad (3.16)$$

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases} \quad (3.17)$$



Obr. 3.23: Exponenciální rozdělení.

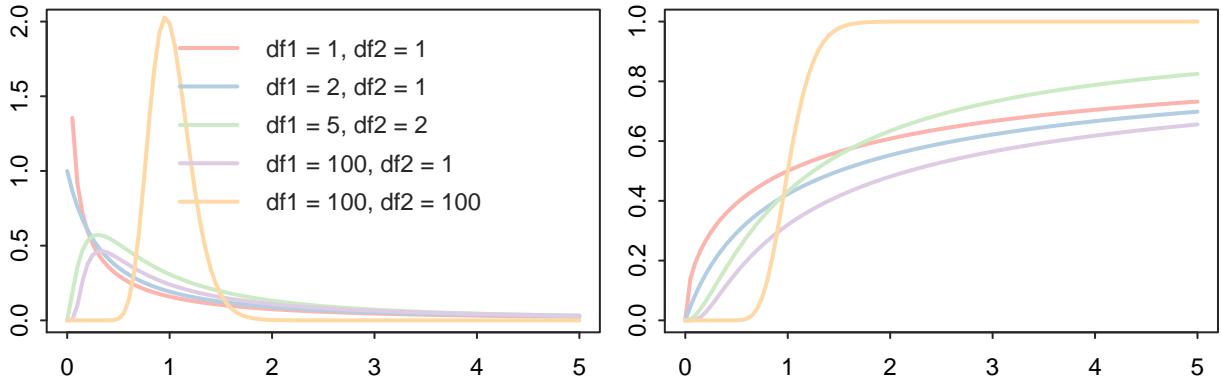
Parametr λ v R jako rate, rodina R funkcí je xexp.

F-rozdělení

$$F(x; df_1, df_2) = I_{\frac{df_1 x}{df_1 x + df_2}} \left(\frac{df_1}{2}, \frac{df_2}{2} \right), \quad (3.18)$$

$$f(x; df_1, df_2) = \frac{1}{B\left(\frac{df_1}{2}, \frac{df_2}{2}\right)} \left(\frac{df_1}{df_2}\right)^{\frac{df_1}{2}} x^{\frac{df_1}{2}-1} \left(1 + \frac{df_1}{df_2}x\right)^{-\frac{df_1+df_2}{2}} \quad (3.19)$$

v R rodina funkcí `xf`.



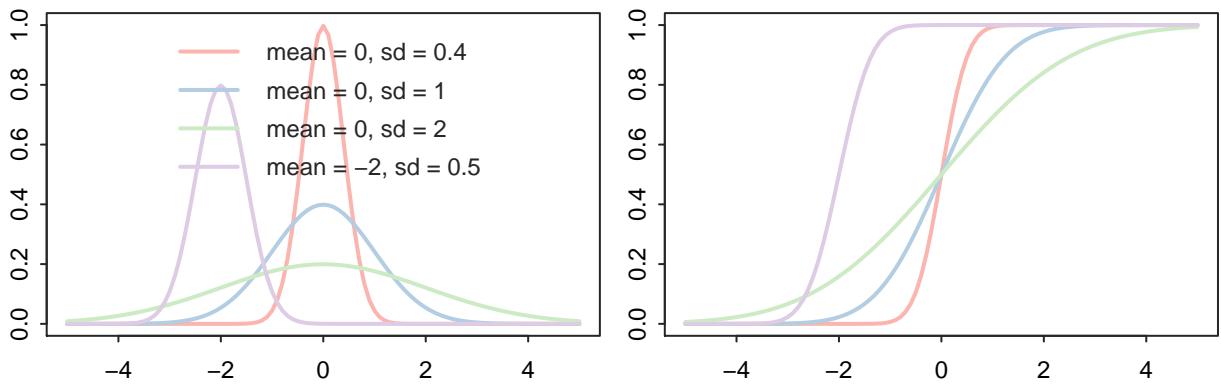
Obr. 3.24: F-rozdělení.

Normální rozdělení

$$F(x) = \frac{1}{\sqrt{\sigma 2\pi}} \int_{-\infty}^x \exp \left[\frac{(x-\mu)^2}{2\sigma^2} \right] dx \quad (3.20)$$

$$f(x, \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[\frac{(x-\mu)^2}{2\sigma^2} \right] \quad (3.21)$$

v R rodina funkcí `xnorm`.



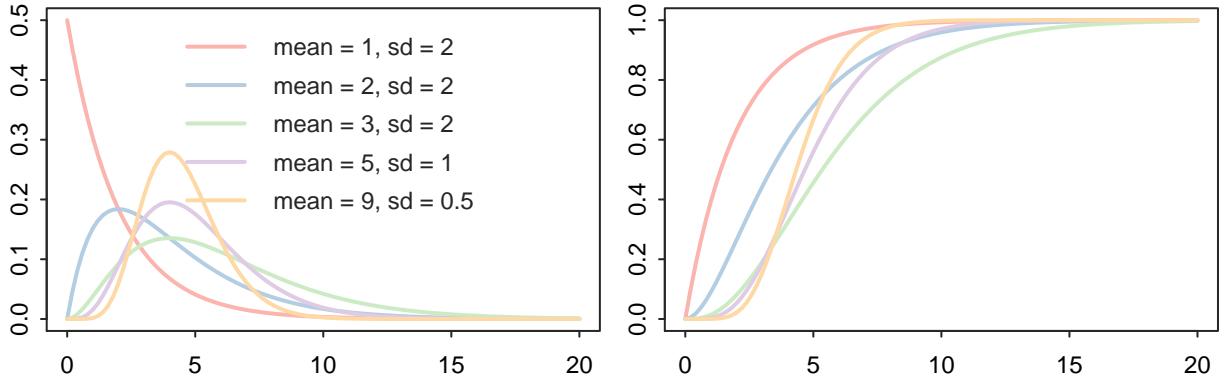
Obr. 3.25: Normální rozdělení.

Gama rozdělení

$$F(x) = \frac{1}{\Gamma(\alpha)} \gamma(\alpha, \beta x) \quad (3.22)$$

$$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \quad (3.23)$$

v R rodina funkcí `xgamma`.



Obr. 3.26: Gama rozdělení.

Studentovo t-rozdělení

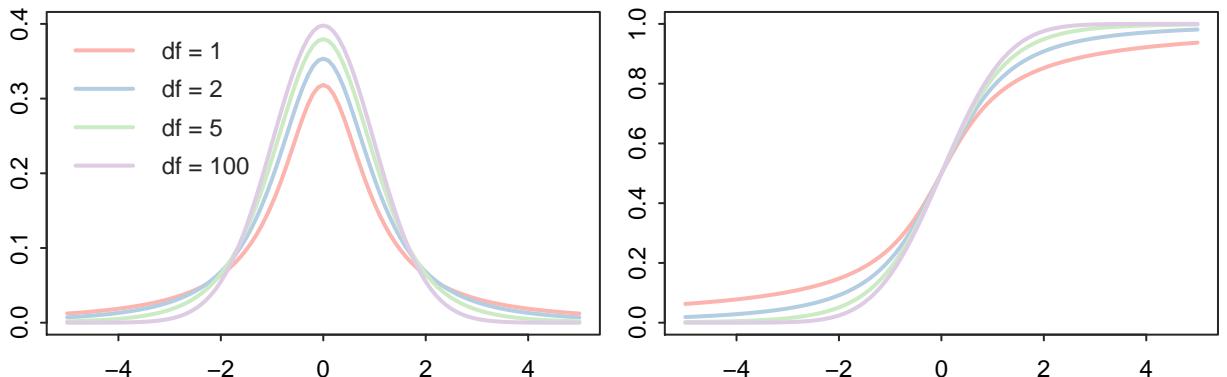
$$F(t) = \int_{-\infty}^t f(u) du = 1 - \frac{1}{2} I_{x(t)}\left(\frac{v}{2}, \frac{1}{2}\right), \quad (3.24)$$

kde I je nekompletní beta funkce a $x(t) = \frac{v}{t^2+v}$.

$$f(t) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\Gamma(\frac{v}{2})} \left(1 + \frac{t^2}{v}\right)^{-\frac{v+1}{2}}, \quad (3.25)$$

kde v je počet stupňů volnosti a Γ je gamma funkce.

v R rodina funkcí `xt`.



Obr. 3.27: Studentovo t-rozdělení.

4 Popisná statistika

Cílem vyhodnocení hydrologických dat je zpravidla jejich sumarizace či charakteristika - např. popis centrální tendence v datech (očekávané hodnoty), odhad doby opakování nějaké hodnoty atp. Jelikož tyto charakteristiky odhadujeme na základě výběru z neznámé veličiny (např. 10 let měření průtoků), jde o výběrové charakteristiky. Ty jsou odhadem parametrů (v principu neznámého) rozdelení náhodné veličiny. Je tedy potřeba rozlišit neznámé charakteristiky zkoumané veličiny a jejich odhad uskutečněný na základě (omezeného) pozorování (výběru) této veličiny. Charakteristiky výběru jsou často jedinou dostupnou informací o zkoumané veličině. Dále se budeme zabývat nejpoužívanějšími charakteristikami výběru $X = (x_1, x_2, \dots, x_i, \dots, x_N)$.

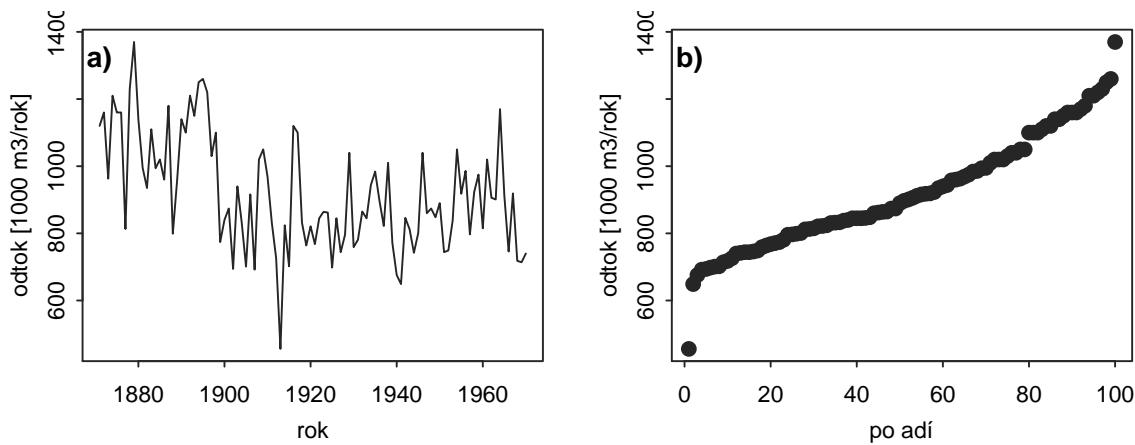
4.1 Empirická distribuční funkce a čára překročení

Základní charakteristikou výběru $X = (x_1, x_2, \dots, x_N)$ je empirická distribuční funkce, kterou získáme na základě uspořádaného výběru

$$x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(k)} \leq \dots \leq x_{(N)} \quad (4.1)$$

kde N je počet prvků výběru a (k) je k -tý prvek dle pořadí od nejmenšího, tzv. ktá pořádková statistika. Například pro Nile data

```
> plot(Nile, ylab='odtok [1000 m3/rok]', xlab = 'rok')
> title(main='a)', line=-1, adj=0.01)
>
> plot(sort(Nile), pch=19, ylab='odtok [1000 m3/rok]', xlab='pořadí')
> title(main='b)', line=-1, adj=0.01)
```



Obr. 4.1: a) Nile data set a b) odpovídající pořádková statistika.

Pořádková statistika tvarem odpovídá distribuční funkci. Je to logické, jelikož pro nějakou měřenou hodnotu, např. pro průtok 984 (tis. m³) víme, že je 67. v pořadí, tj. že pro 67 let ze 100 je průtok stejný

nebo nižší. Zbývá „jen“ přiřadit pravděpodobnosti. Nejzákladnější odhad pravděpodobnosti získáme jako $67/100 = 0.67$, neboli k/N . Tento odhad nicméně má tu nevýhodu, že pro nejvyšší prvek (tj. $k = N$) vychází $F(k) = P(X \leq k) = 1$. Tedy žádnou vyšší hodnotu než maximální z výběru nemůžeme očekávat. To samozřejmě neodpovídá realitě - měříme-li např. 10 let průtok a pozorované maximum z tohoto období bylo $100 \text{ m}^3/\text{s}$ je možné, že budeme pozorovat průtok vyšší než toto maximum. Výpočet pravděpodobnosti je proto nutné provést jiným způsobem. Pravděpodobnost odpovídající které pořádkové statistice se někdy označuje *plotting position* - způsob výpočtu pravděpodobnosti vyjadřuje, kam v grafu umístíme jednotlivé body. Tento výpočet není jednoznačný a v literatuře pro něj existuje řada vztahů. Gumbel (1958) udává pět podmínek, které musí plotting position splňovat:

- 1 Pravděpodobnost pro všechny prvky výběru musí být větší než 0 a menší než 1.
- 2 Pravděpodobnost pro k -tý prvek musí ležet mezi k/N a $(k-1)/N$ (aby byla naplněna podmínka 1) a výpočet pravděpodobnosti nesmí záviset na rozdělení populace, ze které je realizován výběr
- 3 Pravděpodobnost překročení výběrového maxima (nebo nedosažení výběrového minima) by měla být blízkou $1/N$
- 4 Pravděpodobnosti by měly rovnoměrně pokrývat interval $P(X \leq \min(x)) - P(X \leq \max(x))$
- 5 Výpočet pravděpodobností by měl být jednoduchý a intuitivní

V praxi se používá řada vzorců splňujících výše uvedené podmínky, např.

$$P(X \leq x_{(k)}) \approx \frac{k}{N+1} \approx \frac{k-0.5}{N} \approx \frac{k-0.3}{N+0.4} \quad (4.2)$$

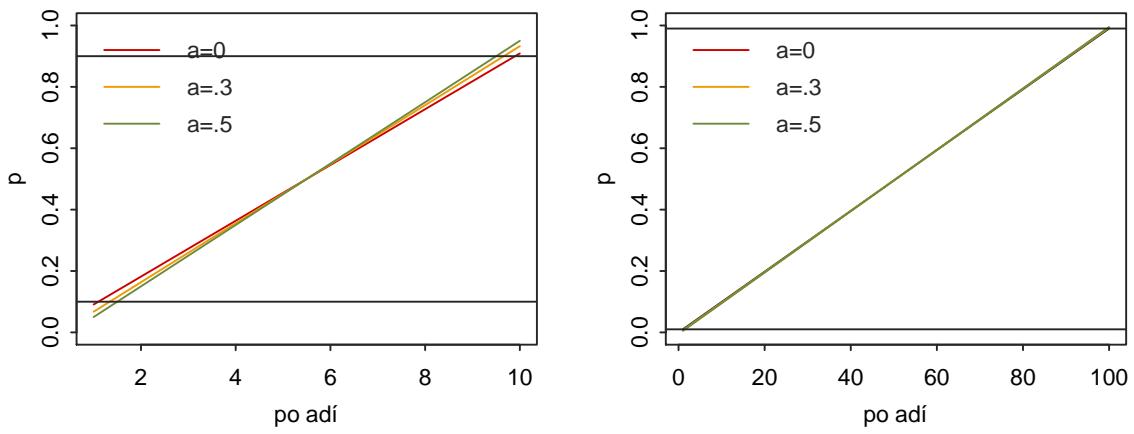
Všechny tyto vzorce je možno schematizovat jako

$$P(X \leq x_{(k)}) \approx \frac{k-a}{N+(1-a)-a} \quad (4.3)$$

výše uvedené případy získáme dosazením $a = 0, 0.5$ a 0.3 .

ÚKOL 4.1 V následujícím příkladu porovnáváme vypočtené pravděpodobnosti pro různá a , konkrétně pro $a = 0, 0.3$ a 0.5 .

```
> p = function(x, a){(rank(x)-a)/(length(x)+(1-a)-a)}
>
> N = 10
> plot(1:N,p(1:N, 0), type='l', ylim = c(0, 1), xlab='pořadí', ylab='p', col='red3')
> lines(1:N,p(1:N, .3), col='orange2')
> lines(1:N,p(1:N, .5), col='darkolivegreen4')
> abline(h=c(1, N-1)/N)
> legend('topleft', c('a=0', 'a=.3', 'a=.5'), col=c('red3', 'orange2', 'darkolivegreen4'), lty=1, bty='n')
>
> N = 100
> plot(1:N,p(1:N, 0), type='l', ylim = c(0, 1), xlab='pořadí', ylab='p')
> lines(1:N,p(1:N, .3), col='orange2')
> lines(1:N,p(1:N, .5), col='darkolivegreen4')
> abline(h=c(1, N-1)/N)
> legend('topleft', c('a=0', 'a=.3', 'a=.5'), col=c('red3', 'orange2', 'darkolivegreen4'), lty=1, bty='n')
```

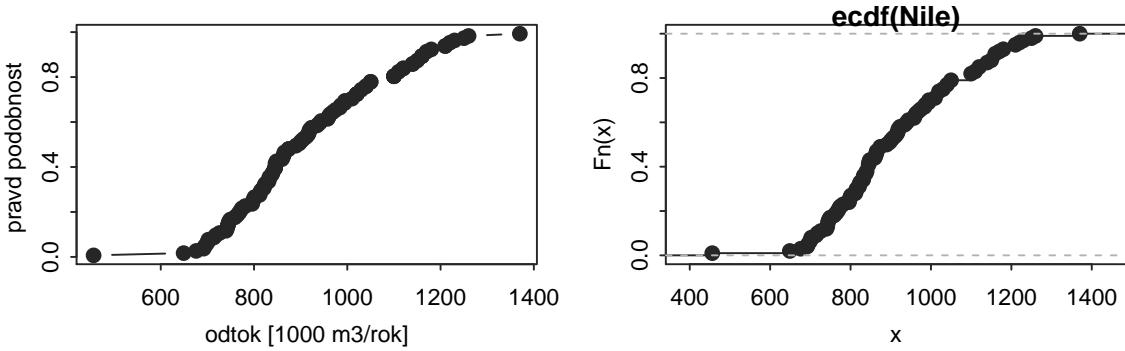


Obr. 4.2: Porovnání různých plotting positions pro $N = 10$ (vlevo) a $N = 100$ (vpravo).

□

Je evidentní, že význam způsobu výpočtu pravděpodobnosti může mít velký vliv zejména v případě malých výběrů a extrémů. V R je možno empirickou distribuční funkci vykreslit buď na základě výpočtu pravděpodobností, nebo pomocí funkce `ecdf`:

```
> p = function(x, a){(rank(x)-a)/(length(x)+(1-a)-a)}
>
> plot(sort(Nile), p(sort(Nile), a=.3), type='b', ylab='pravděpodobnost', xlab='odtok [1000 m3/rok]', pch=19)
>
> plot(ecdf(Nile))
```



Obr. 4.3: Ukázka odvození empirické distribuční funkce - „ručně“ (vlevo), pomocí funkce `ecdf` (vpravo).

Funkce `ecdf` navíc vrací distribuční funkci. Např. pravděpodobnost, že průtok je menší nebo roven 800 tis. m^3 je možno spočítat pomocí

```
> dfce = ecdf(Nile)
> dfce(800)
[1] 0.26
```

nebo stručněji

```
> ecdf(Nile)(800)
[1] 0.26
```

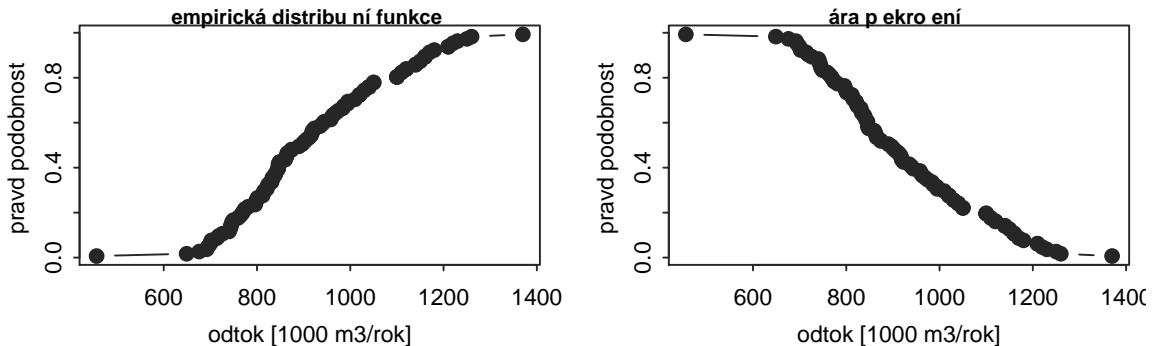
Čára překročení

V hydrologii se často místo empirické distribuční funkce používá čára překročení. Ta je založena na sestupně uspořádaném výběru

$$x_{(1)} > x_{(2)} > \dots > x_{(k)} > \dots > x_{(N)} \quad (4.4)$$

k jednotlivým pořádkovým statistikám jsou pak přiřazeny pravděpodobnosti, že veličina nabývá hodnot vyšších než $x_{(k)}$, tedy $P(X > x_{(k)})$ podobně jako v případě vzestupné pořádkové statistiky. Přitom platí, že $P(X > x) = 1 - P(X \leq x)$. Čára překročení je pouze jiným vyjádřením empirické distribuční funkce.

```
> p = function(x, a){(rank(x)-a)/(length(x)+(1-a)-a)}
>
> plot(sort(Nile), p(sort(Nile), a=.3), type='b', ylab='pravděpodobnost', xlab='odtok [1000 m3/rok]', pch=19,
> plot(sort(Nile), 1-p(sort(Nile), a=.3), type='b', ylab='pravděpodobnost', xlab='odtok [1000 m3/rok]', pch=19,
```



Obr. 4.4: Empirická distribuční funkce (vlevo) a čára překročení (vpravo).

4.2 Výběrové kvantily

Výběrový $100p$ -procentní kvantil odhadneme tak, že vypočteme

$$1 \quad k.d = (n+1-2a)p+a,$$

$$2 \quad q_p = x_{(k)} + d(x_{(k+1)} - x_{(k)})$$

kde a souvisí s volbou plotting position (viz rovnice 4.3).

ÚKOL 4.2 Spočítejte 10% kvantil ročních průtoků Nilu.

```
> p = 0.1
> n = length(Nile) # počet pozorování
> a = 0 # zvolíme dle uvážení
> k.d = (n+1-2*a)*p+a
> k.d
```

```
[1] 10.1
```

v proměnné $k.d$ označuje k celočíselnou část a d desetinnou část, tedy

```
> k = floor(k.d)
> k
[1] 10
> d = k.d - k
> d
[1] 0.1
```

výsledek je pak získán lineární interpolací mezi 10 a 11 hodnotou dle výše uvedeného vzorce, tedy

```
> ordNile = sort(Nile) # vytvoříme pořádkovou statistiku
> q10 = ordNile[k] + d * (ordNile[k+1] - ordNile[k])
> q10
[1] 718.8
```

□

V R získáme kvantily pomocí funkce `quantile`. Tedy 10% kvantil ročních průtoků Nilu

```
> quantile(Nile, probs = 0.1)
10%
725.2
```

Všimněte si, že výsledek se liší od hodnoty uvedené v příkladu 4.2. Důvodem je jiná hodnota konstanty a . Tato konstanta bohužel nejde přímo zadávat do funkce `quantile`, nicméně R zná 9 typů výpočtu kvantilů (respektive pravděpodobnosti dle pořadí), které se zadávají pomocí parametru `type`. Jednotlivé typy lze popsat následovně:

type 1 pokud $k = np$ je celé číslo, pak $q_p = x_{(k)}$, jinak $q_p = x_{(k+1)}$

type 2 pokud $k = np$ je celé číslo, pak $q_p = x_{(k)}$, jinak $q_p = 0.5x_{(k)} + 0.5x_{(k+1)}$

type 3 nejbližší lichá pořádková statistika

type 4 $k.d = np$

type 5 $k.d = np + 0.5$

type 6 $k.d = (n+1)p$

type 7 $k.d = (n-1)p + 1$

type 8 $k.d = (n+1/3)p + 1/3$

type 9 $k.d = (n+1/4) + 3/8$

Defaultně Rko používá `type=7`, výsledek v příkladu odpovídá typu 6, tedy

```
> quantile(Nile, probs = 0.1, type=6)
10%
718.8
```

Přehled odhadů pro jednotlivé typy výpočtu pro Nile data:

```
> q10=mapply(quantile, type=1:9, MoreArgs = list(probs=0.1, x=Nile))
> names(q10) = paste('type', 1:9)
> q10
```

```

type 1 type 2 type 3 type 4 type 5 type 6 type 7 type 8 type 9
718.0 722.0 718.0 718.0 722.0 718.8 725.2 720.9 721.2

```

V hydrologii se asi nejčastěji používá typ 6, případně varianta s $a = 0.3$, která zhruba odpovídá typu 8.

4.3 Výběrové momenty

Podobně jako v případě náhodných veličin můžeme i výběr charakterizovat pomocí (výběrových) momentů.

Obecné momenty řádu k je možno odhadnout jako

$$m_k = \frac{1}{df} \sum_i x_i^k \quad (4.5)$$

Centrální momenty pak

$$c_k = \frac{1}{df} \sum_i (x_i - m_1)^k \quad (4.6)$$

Ve jmenovateli rovnic 4.5 a 4.6 se neuvažuje počet prvků výběru, ale tzv. stupně volnosti df , tj. počet prvků výběru zmenšený o počet vazeb, čili o počet odhadovaných parametrů. Pro obecné momenty je počet stupňů volnosti roven počtu prvků výběru N , pro centrální momenty je počet stupňů volnosti roven počtu prvků minus 1, kvůli tomu, že ve výpočtu figuruje odhadovaný první obecný moment (průměr), tedy $df = N - 1$.

4.4 Míry polohy výběru

Pythagorejské průměry

Poloha výběru je základní charakteristika. Nejčastějším ukazatelem je aritmetický průměr

$$a = \bar{x} = \frac{1}{N} \sum_i x_i \quad (4.7)$$

někdy je výhodné zjistit geometrický průměr

$$g = \left(\prod_i x_i \right)^{1/N} = \sqrt[N]{x_1 x_2 \cdots x_N}. \quad (4.8)$$

nebo harmonický průměr

$$h = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_N}} = \frac{N}{\sum_i \frac{1}{x_i}} \quad (4.9)$$

Pro tyto tři průměry platí, že $h \leq g \leq a$. Geometrický průměr se často používá pro výběry pocházející z asymetrických rozdělení.

Modus, medián

Nejpravděpodobnější hodnota z výběru se nazývá modus *mode*. Většina rozdělení, se kterými se setkáváme v hydrologii, je unimodální - tj. hustota pravděpodobnosti má pouze globální maximum. Nicméně v případě, že data jsou generována různými procesy, může být rozdělení výběru i vícemodální.

Další hojně využívanou charakteristikou je medián, tj. 50% výběrový kvantil. Tato charakteristika se řadí mezi tzv. robustní odhady, jelikož není ovlivněna odlehlými hodnotami (závisí pouze na poloze „prostředních“ pozorování). Pokud je velikost výběru liché číslo, medián \tilde{x} se nachází na $(N+1)/2$ pozici seřazeného výběru. Pokud je velikost výběru sudé číslo, \tilde{x} se spočítá jako průměr prvků na $N/2$ a $(N+1)/2$ pozici seřazené řady.

Pro unimodální rozdělení platí, že

$$\frac{|\tilde{x} - \bar{x}|}{s_x} \leq (3/5)^{1/2} \quad (4.10)$$

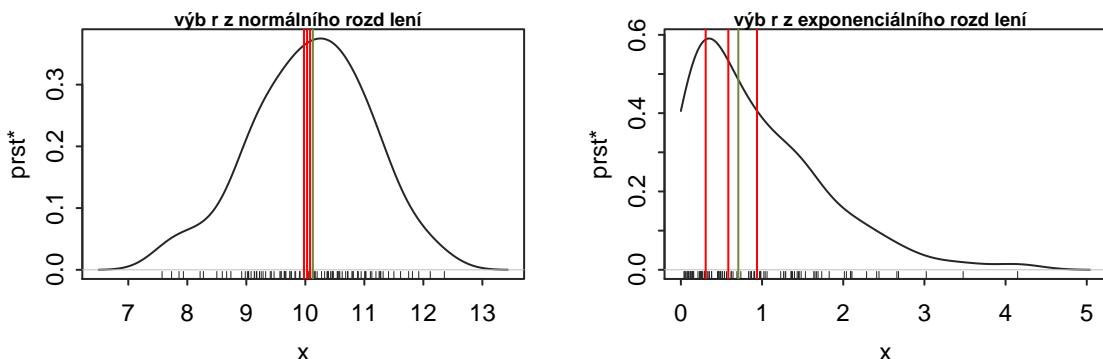
a

$$\frac{|\tilde{x} - \text{mode}(x)|}{s_x} \leq 3^{1/2}, \quad (4.11)$$

kde s_x je směrodatná odchylka, definovaná níže. Zároveň platí, že pro symetrická rozdělení $\bar{x} = \tilde{x} = \text{mode}(x)$, pro rozdělení zešikmená zprava platí, že $\bar{x} > \tilde{x} > \text{mode}(x)$ a pro rozdělení zešikmená zleva naopak.

ÚKOL 4.3 Porovnejte odhady průměru a mediánu pro výběr ze symetrického a asymetrického rozdělení. Identifikujte aritmetický, geometrický a harmonický průměr.

```
> x = rnorm(100, mean=10)
> plot(density(x), main = 'výběr z normálního rozdělení', cex.main=.8, ylab='prst*', xlab='x')
> rug(x)
> abline(v=c(mean(x), prod(x)^(1/length(x)), length(x) / sum(1/x)), col='red')
> abline(v=median(x), col='darkolivegreen4')
>
> x = rexp(100)
> plot(density(x, from=0), main = 'výběr z exponenciálního rozdělení', cex.main=.8, xlab='x', ylab='prst')
> rug(x)
> abline(v=c(mean(x), prod(x)^(1/length(x)), length(x) / sum(1/x)), col='red')
> abline(v=median(x), col='darkolivegreen4')
```



Obr. 4.5: Porovnání mediánu (zeleně) a aritmetického, geometrického a harmonického průměru pro dvě rozdělení. Černá čára ukazuje odhad hustoty rozdělení.

□

Rko zná funkce `mean` pro výpočet výběrového průměru a `median` pro výpočet výběrového mediánu. Funkce `mode`, která v R existuje se netýká modu rozdělení.

4.5 Míry variability

Základní mírou variability výběru je výběrový rozptyl

$$s_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4.12)$$

a výběrová směrodatná odchylka

$$s_x = \sqrt{s_x^2} \quad (4.13)$$

tedy průměrná odchylka od průměru a zároveň druhý centrální výběrový moment.

Pro data nabývající pouze nezáporných hodnot, jako např. srážky a průtoky, se v praxi často uvažuje koeficient variace (relativní variabilita) definovaný jako

$$cv = \frac{s_x}{\bar{x}} \quad (4.14)$$

Jelikož tyto charakteristiky jsou ovlivněny odlehlými hodnotami, používají se i robustnější alternativy:

IQR - mezikvartilové rozpětí, tj.

$$IQR = q_{0.75} - q_{0.25} \quad (4.15)$$

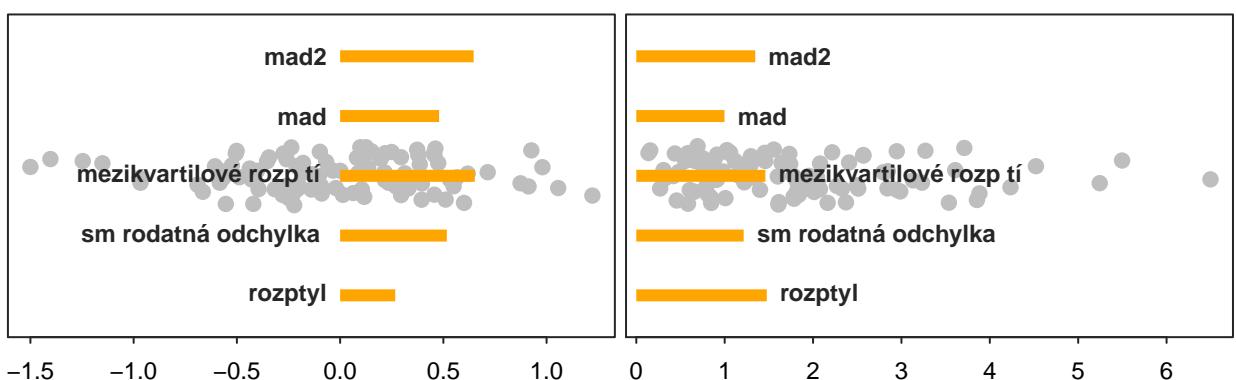
mad - mediánová absolutní odchylka (*median absolute deviation*)

$$mad \propto \text{median}(|x_1 - \tilde{x}|, |x_2 - \tilde{x}|, \dots, |x_N - \tilde{x}|), \text{tj.} \quad (4.16)$$

$$mad_c = c \cdot \text{median}(|x_1 - \tilde{x}|, |x_2 - \tilde{x}|, \dots, |x_N - \tilde{x}|) \quad (4.17)$$

kde \tilde{x} je výběrový medián a c je měřítková konstanta, pokud má X normální rozdělení, je $c = 1.4286$.

Dále je někdy užitečné zjistit rozpětí výběru (rozdíl maxima a minima). Porovnání jednotlivých charakteristik variability výběru ukazuje obrázek 4.6.



Obr. 4.6: Porovnání charakteristik variability pro výběr z normálního (vlevo) a gama (vpravo) rozdělení. Šedé body zobrazují výběr.

Rko zná funkce `sd` pro výpočet směrodatné odchyly, `var` pro výpočet rozptylu, `IQR` pro výpočet mezikvartilového rozpětí a `mad` pro výpočet mediánové absolutní odchylky. Koeficient variace musí být spočítán, a to jako `sd(x) / mean(x)`. Minimální a maximální hodnotu lze zjistit souhrnně pomocí `range`, rozpětí (velikost intervalu mezi maximem a minimem) pomocí `diff(range(X))`.

4.6 Míry tvaru

Výběrová šikmost

$$c_3 = \frac{1}{N} \frac{\sum_i (x_i - \bar{x})^3}{s_X^3} \quad (4.18)$$

Pro výběrovou šikmost platí

- $-\infty < c_3 < \infty$
- $c_3 < 0$ pro zprava zešikmené rozdělení
- $c_3 > 0$ pro zleva zešikmené rozdělení
- $c_3 = 0$ pro symetrické rozdělení
- rozdělení je významně šikmé, pokud $|c_3| > 2\sqrt{6/N}$

Výběrová špičatost

$$c_4 = \frac{1}{N} \frac{\sum_i (x_i - \bar{x})^4}{s_X^4} - 3 \quad (4.19)$$

a platí

- $-2 < c_4 < \infty$

4.7 Histogram

Prakticky není zcela jednoduché odhadnout hustotu rozdělení na základě výběru. Pro charakterizaci výběru je proto spíše využíván histogram. Histogram vyjadřuje četnost pozorování v předem definovaných třídních intervalech. Postup tvorby histogramu pro výběr X lze shrnout následovně:

- 1 Zvolte počet tříd četnosti.
- 2 Rozdělte rozpětí dat (`range(X)`) na tento počet intervalů.
- 3 Spočítejte, kolik hodnot z X připadá do jednotlivých intervalů.
- 4 Vykreslete graf s třídními intervaly na ose x a četnostmi na ose y. Někdy se vykreslují relativní četnosti - tj. počet prvků v intervalu děleno celkový počet prvků v X .

Pro odhad počtu tříd četnosti existuje řada vztahů. V základním R jsou implementovány tři:

Sturges

$$\lceil \log_2(n) + 1 \rceil \quad (4.20)$$

Scott

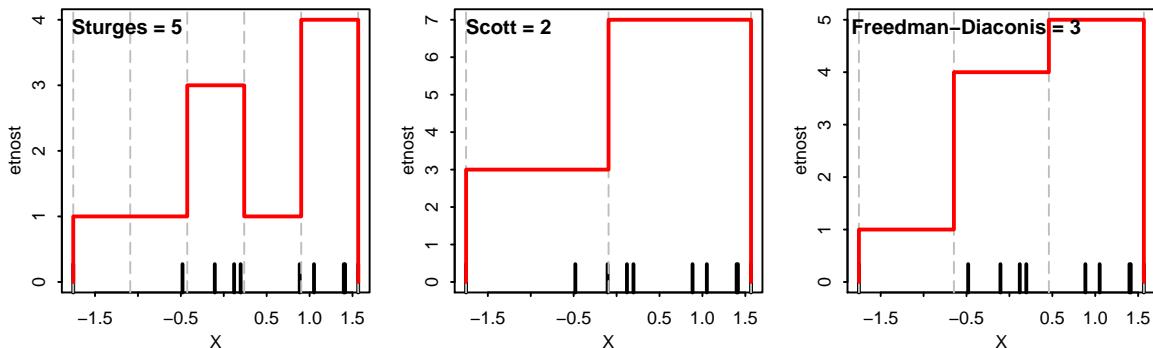
$$\left\lceil \sqrt[3]{n} \frac{\min(X) - \max(X)}{3.5s_X} \right\rceil \quad \text{pro } s_X > 0, \quad \text{jinak 1} \quad (4.21)$$

Freedman-Diaconis

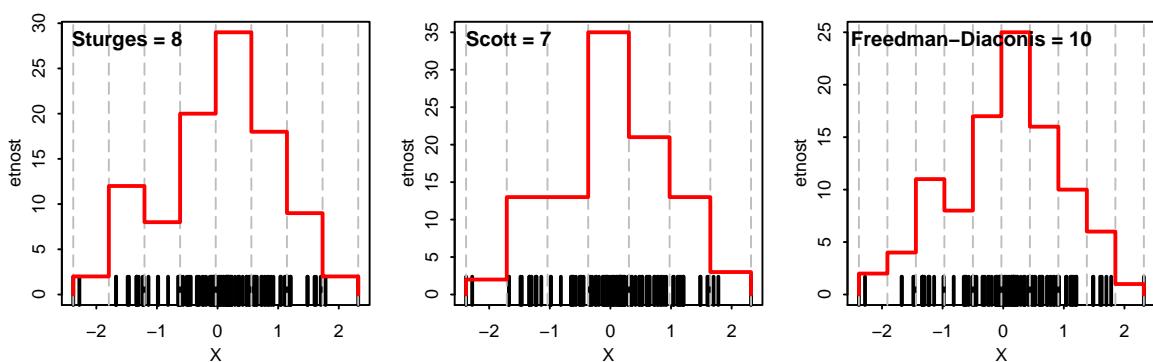
$$\left\lceil \sqrt[3]{n} \frac{\min(X) - \max(X)}{2 \text{IQR}(X)} \right\rceil \quad \text{pro } \text{IQR}(X) > 0, \quad \text{jinak } \text{mad}_2(X), \\ \text{pokud i } \text{mad}_2(X) = 0, \text{ pak 1} \quad (4.22)$$

kde $\lceil x \rceil$ znamená nejbližší vyšší celé číslo k číslu x . Tyto odhady jsou v R implementovány pomocí funkcí `nclass.Sturges`, `nclas.scott` a `nclass.FD`.

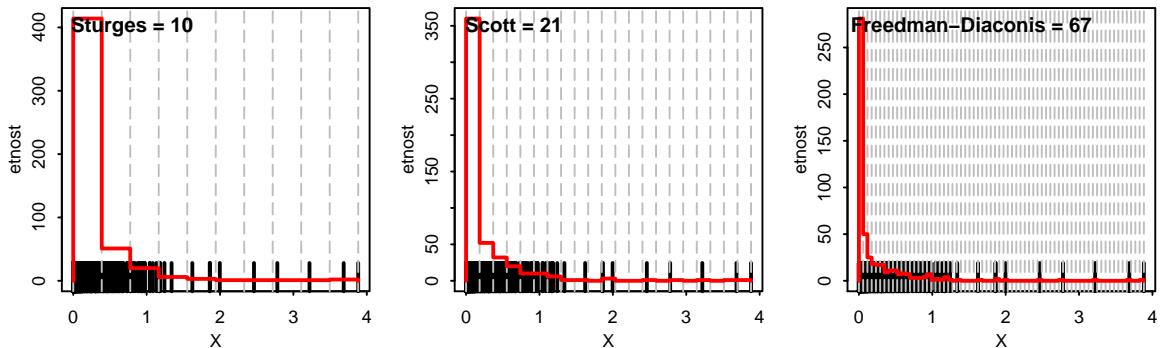
Aby bylo vůbec možné vykreslit histogram, potřebujeme mít k dispozici dostatečně velký výběr, respektive dostatečně pokryté rozpětí výběru. V závislosti na tom volíme i vhodnou metodu odhadu počtu tříd četnosti. Na obrázku 4.7 jsou vytvořeny histogramy pro různé třídní intervaly. Počet tříd dle Sturgese je nejvyšší, proto se v histogramu objevují poměrně často intervaly s nízkými četnostmi, jako vhodnější se v tomto případě jeví odhad dle Scotta nebo Freedmana-Diaconise. Rozdíly jsou méně patrné, pokud je výběr větší (viz obr. 4.8). Nicméně v případě asymetricky rozdělených výběrů se odhad počtu tříd četnosti značně liší i pro relativně velké výběry (viz obr. 4.9).



Obr. 4.7: Histogramy pro 10prvkový výběr. Červená čára zobrazuje histogram, šedě jsou vyznačeny třídy četnosti a černě data z výběru.



Obr. 4.8: Histogramy pro 100prvkový výběr. Červená čára zobrazuje histogram, šedě jsou vyznačeny třídy četnosti a černě data z výběru.



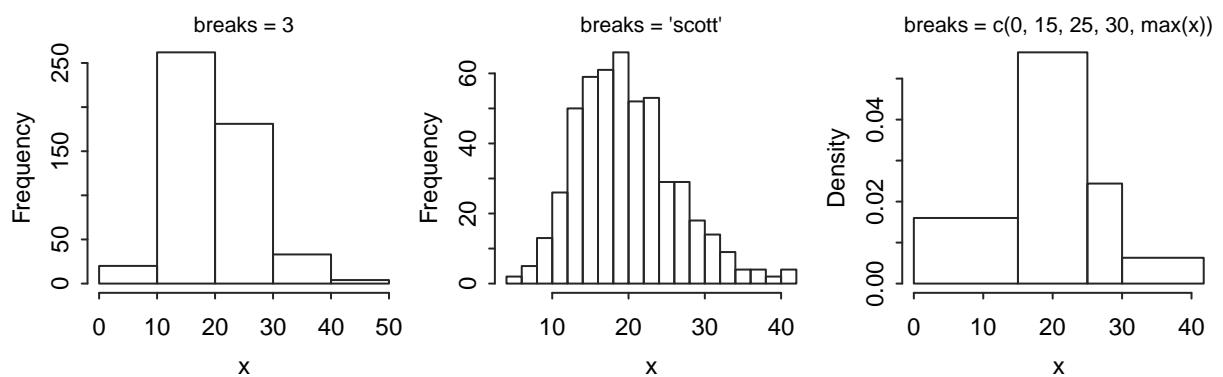
Obr. 4.9: Histogramy pro 100prvkový výběr z asymetrického rozdělení. Červená čára zobrazuje histogram, šedě jsou vyznačeny třídy četnosti a černě data z výběru.

Histogramy v R je možno vytvořit pomocí funkce `hist`. Funkce `hist` má řadu argumentů:

```
> args(hist.default)
function (x, breaks = "Sturges", freq = NULL, probability = !freq,
  include.lowest = TRUE, right = TRUE, density = NULL, angle = 45,
  col = NULL, border = NULL, main = paste("Histogram of", xname),
  xlim = range(breaks), ylim = NULL, xlab = xname, ylab, axes = TRUE,
  plot = TRUE, labels = FALSE, nclass = NULL, warn.unused = TRUE,
  ...)
NULL
```

Podstatný je argument `breaks`, který udává buď preferovaný počet tříd, metodu výpočtu tříd, nebo přímo hranice tříd četnosti:

```
> x= rchisq(500, df=20)
> hist(x, breaks = 3, main='breaks = 3')
> hist(x, breaks = 'scott', main="breaks = 'scott'")
> hist(x, breaks = c(0, 15, 25, 30, max(x)), main="breaks = c(0, 15, 25, 30, max(x))")
```



Obr. 4.10: Různé způsoby zadání parametru `breaks` ve funkci `hist`.

Všimněte si, že pokud zadáme různě velké třídy četnosti, R automaticky změní zobrazení z četnosti (*frequency*) na relativní četnost. Toto chování je jinak možné ovlivňovat nastavením parametrů `freq` nebo `probability`. Pokud nastavíme parametr `breaks` na konkrétní číslo, R se snaží najít „pěkné“

hranice tříd pomocí funkce `pretty`, důsledkem může být mírně odlišný počet tříd.

4.8 Krabicový graf - *Boxplot*

Jednou z možností jak summarizovat rozdělení dat je tzv. pětičíselná statistika (*five number summary*) - $5NS$, která je definována jako

$$5NS(X) = (x_{(1)}, h_L, \tilde{x}, h_U, x_{(N)}), \quad (4.23)$$

kde

- h_L je dolní kvartil (tj. $x_{0.25}$)
- h_U je horní kvartil (tj. $x_{0.75}$)
- \tilde{x} je medián
- zpravidla jsou ještě separovány odlehlé hodnoty (*outlier*) - za potenciálně odlehlé hodnoty jsou označena taková pozorování x , pro která platí, že

$$x < h_L - 1.5 IQR(X), \quad \text{nebo} \quad (4.24)$$

$$x > h_U + 1.5 IQR(X) \quad (4.25)$$

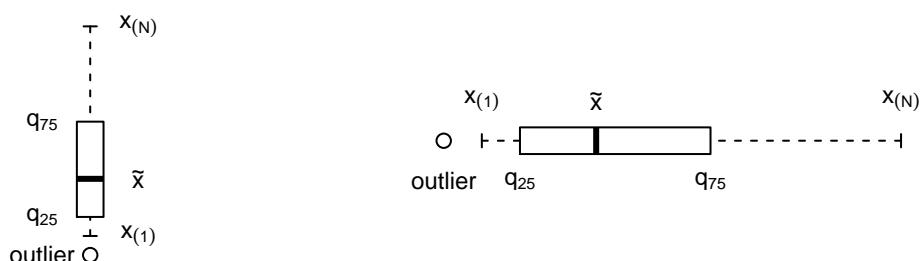
- za $x_{(1)}$, resp. $x_{(N)}$ je poté dosazena nejmenší (nejvyšší) hodnota z výběru bez potenciálně odlehlých hodnot

ÚKOL 4.4 Jaká je pravděpodobnost, že $x < h_L - 1.5 IQR(X)$, respektive $x > h_U + 1.5 IQR(X)$, pokud je x z normálního rozdělení? Záleží na parametrech rozdělení a na samotném rozdělení?

```
> hL = qnorm(.25)
> hU = qnorm(.75)
> iqr = hU - hL
> pnorm(hL - 1.5 * iqr)
[1] 0.003488
> 1-pnorm(hU + 1.5 * iqr)
[1] 0.003488
```

□

Grafickým znázorněním pětičíselné statistiky je boxplot (krabicový graf), viz obr. 4.11. „Krabice“ znázorňuje mezikvartilové rozpětí, „vousy“ (whiskers) zobrazují standardně $x_{(1)}$ a $x_{(N)}$ bez odlehlých hodnot, nicméně v některých aplikacích se používá např. i 5 a 95 % kvantil apod.



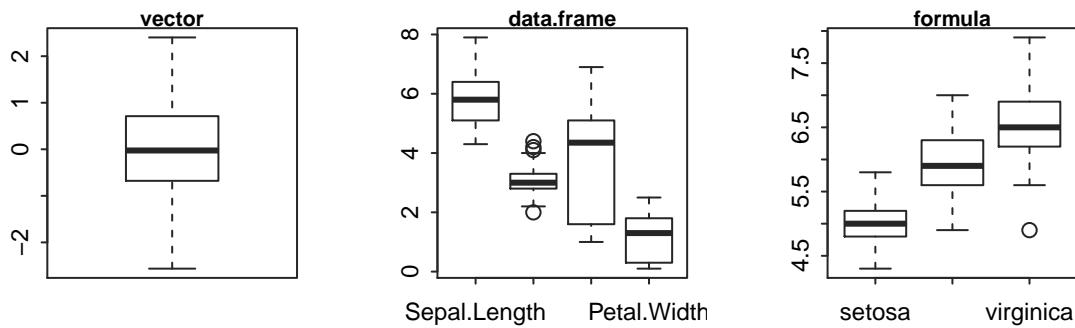
Obr. 4.11: Mapování pětičíselné statistiky na boxplot.

V Rku lze boxplot vytvářet mnoha způsoby, přehled nejpoužívanějších udávají následující kódy.

```
> # boxplot pro vektor
> boxplot(rnorm(100), main='vector')
>
> data(iris) # načti data.frame iris
> head(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5         1.4         0.2   setosa
2          4.9         3.0         1.4         0.2   setosa
3          4.7         3.2         1.3         0.2   setosa
4          4.6         3.1         1.5         0.2   setosa
5          5.0         3.6         1.4         0.2   setosa
6          5.4         3.9         1.7         0.4   setosa

> # boxplot pro data.frame
> boxplot(iris[, 1:4], main='data.frame', horizontal=FALSE)
> # boxplot pro třídu formula - zahrnující více veličin
> boxplot(iris$Sepal.Length~iris$Species, main='formula')
```



Obr. 4.12: Ukázka tvorby krabicových grafů.

5 Vodohospodářská data a jejich vizualizace

Klimatický systém zahrnuje nejen atmosféru, ale také složky hydrosféry a biosféru. Jednotlivé složky jsou propojeny pomocí *hydrologického cyklu*. V této kapitole se budeme zabývat rozdelením dat, které ovlivňují vodní hospodářství, a jejich grafickou interpretací.

5.1 Typy dat

Obecně lze data rozdělit na:

1 Kvantitativní data:

- míra nebo množství něčeho
- číselné hodnoty
- výsledky měření
- diskrétní
 - spočtená množina hodnot
 - počet dnů se srážkami
- spojitá
 - libovolné hodnoty z intervalu
 - výška srážky

2 Kvalitativní data:

- nejsou číselné hodnoty (nejde s nimi počítat)
- faktor - podmínka pro rozdelení dat do kategorií
- úrovně faktoru - možné hodnoty
- faktor
 - nominální - název pozorovací stanice
 - ordinální (lze stanovit pořadí)
 - stupeň povodňové aktivity (I., II., III.)

3 Logická data:

- dvouhodnotová: pravda, nepravda (v R TRUE, FALSE)

4 Chybějící data: v prostředí R NA

5 Další...

Dále je nutné u dat rozlišovat jejich prostorové a časové měřítko, neboli časoprostorové rozlišení dat. Prostorové měřítko lze charakterizovat z hlediska velikosti území, které je v daném okamžiku ovlivněno daným procesem. Analogicky se dá interpretovat časové měřítko. Oba dva jevy jsou navzájem propojeny a dle jejich charakteristik, jako je například doba trvání či velikost zasaženého území, daný jev zařadit (kategorizovat). Často se také krátkodobé události týkají malého území a naopak.

5.1.1 Atmosférické proměnné

Mezi atmosférické proměnné řadíme rychlosť a smer vetrov, slnečný záření atd., avšak dvě proměnné jsou pro dopad na vodní hospodářství přeci jenom důležitější, a to teplota vzduchu a srážkové úhrny.

Teplota vzduchu

Teplota je základní fyzikální veličinou soustavy SI s jednotkou kelvin (K) a vedlejší jednotkou stupeň Celsia ($^{\circ}\text{C}$). Nejnižší možnou teplotou je teplota absolutní nuly (0 K; $-273,15\text{ }^{\circ}\text{C}$), ke které se lze přiblížit, avšak nelze jí dosáhnout. K měření teploty se používají teploměry.

Teplota vzduchu se měří ve výšce 2 metry nad zemským povrchem ve stínu (v meteorologické budce). Zpravidla se udává:

- denní minimální teplota,
- denní maximální teplota a
- průměrná denní teplota (jedná se o aritmetický průměr z teploty vzduchu naměřené v 7 hodin, teploty ve 14 hodin a dvojnásobně započtené teploty v 21 hodin, vše místního středního slunečního času).

Vedle toho se také sleduje přízemní minimální teplota (minimum naměřené za noc ve výšce 5 cm nad zemským povrchem), rosný bod (teplota, při které dosáhne vzduch maximální možné vlhkosti) v různých výškách. K základním klimatickým údajům patří roční průběhy maximální a minimální teploty vzduchu a maximální a minimální povrchové teploty moří a oceánů.

Srážkové úhrny

Atmosférické srážky jsou vodní kapky nebo ledové částice vznikající následkem kondenzace nebo desublimace vodní páry. Ke kondenzaci vodní páry dochází zpravidla při stoupání vzduchu, při němž vzduch expanduje, jelikož atmosférický tlak s výškou klesá. Při rozpínání se vzduch ochlazuje a může docházet ke kondenzaci vodní páry. K ochlazování vzduchu dochází rovněž v důsledku snížení příslušné energie (např. v noci je vzduch ochlazován zemským povrchem - může vznikat mlha či rosa). Srážky se zpravidla dělí na vertikální (např. dešť, sníh, mrholení, kroupy) a horizontální (např. rosa, jinovatka, námraza). Množství srážkového úhrnu se měří pomocí srážkoměrů a je udáváno v milimetrech za časovou jednotku.

Dle způsobu vzniku můžeme vertikální srážky rozdělit na:

- konvektivní srážky - část nestabilní atmosféry je ohřívána více než její okolí, dochází k výraznému vertikálnímu proudění,
- orografické srážky - vzduch je zvedán při přechodu orografické překážky,
- frontální srážky - při pohybu front může docházet k zdvihu teplého vzduchu nad nasouvající se studenou frontou nebo k zvedání teplého vzduchu při nasouvání teplé fronty na frontu studenou.

Kondenzace probíhá na kondenzačních jádrech (např. prachové částice, aerosoly). Pokud jsou kapky uvnitř oblaku dostatečně velké, dochází k vypadávání srážky.

Základní parametry srážky:

- Srážková výška H_s [mm] – výška vodního sloupce, která by se vytvořila z deště na dané ploše bez odtoku, výparu či vsaku.

- Srážkový úhrn – množství srážek vypadlé v bodě (srážkoměrné stanici) vyjádřené rovněž jako výška vodního sloupce.

5.1.2 Hydrologická data

Odtok je nevsáknutá část srážky a vyvěrající voda z podzemních pramenů stékající působením gravitace ve směru největšího sklonu. Plošný odtok je postupné soustředování vody (ron, stružky, potoky, řeky). Hydrologické bilanční složky dělíme na:

1 Odtok

- Průtok Q – objem vody proteklý profilem za jednotku času [$m^3.s^{-1}$]
- Proteklé množství O – objem vody proteklý profilem za delší časové období [$tis.m^3$]
- Denní odtok [m^3, mm]
- Odtok za průměrný měsíc [m^3, mm]
- Odtok za rok
- Odtok za průměrný rok Q_a
- Q_d, Q_m, Q_r – průměrný denní, měsíční, respektive roční průtok
- Specifický průtok, odtok q [$m^3.s^{-1}.km^2$]
- Přirozený průtok - pozorovaný průtok korigovaný dle údajů o umělých regulacích a užívání vody
- Vodní stav [cm]

2 Výpar

- Výpar z volné vodní hladiny
- Výpar ze sněhu a ledu (sublimace)
- Výpar z povrchu půdy (bez vegetace)
- Evapotranspirace
 - Transpirace – voda vydechovaná rostlinami a živočichy do atmosféry
 - Evaporace

3 Zásoby vody

- Zásoba vody ve sněhové pokrývce [m^3, mm]
- Zásoba vody v půdě [m^3, mm]
- Perkolace [m^3, mm]
- Dotace zásob podzemní vody [m^3, mm]

Vodohospodářská data

Vodohospodářská data můžeme rozdělit na:

- Data o užívání vody - dle vyhlášky
- Manipulace na nádržích

Geomorfologická data

- Digitální model terénu
- CORINE - GIS data o užívání půdy
- Říční soustava – hlavní tok se svými přítoky.
- Říční síť – systém říčních soustav.

Charakteristiky toku:

- Pramen – počátek toku – pramen soustředěný či nesoustředěný
- Ústí toku – místo, kde se tok vlévá do jiného toku
- Délka toku L – vzdálenost od pramene k ústí, měřeno osou koryta
- Staničení profilu – vzdálenost daného profilu od ústí, měřeno osou
- Stupeň vývinu toku – d/L , d je délka přímé spojnice pramene a ústí
- Schematický podélný profil

ÚKOL 5.1 Vykreslete graf teplot vzduchu a srážkových úhrnů - využijte data z projektu

5.2 Vizualizace a typy grafů

5.2.1 Vizualizace dat

Vizualizace je tvorba grafické reprezentace dat pro jejich pochopení, nebo jinak vizualizace je tvorba mentální reprezentace dat pomocí grafiky.

Cílem vizualizace je usnadnit porovnávání dat, rozeznání vzorců a detekci změn v datech. Dalším cílem je zpřístupnit komplikované sady dat lidskému vnímání.

5.2.2 Typy grafů

Vizualizovat data je možné mnoha způsoby. Podstatou vizualizace je mapování vybrané proměnné (spojité či diskrétní, často vícerozměrné) na tzv. estetické atributy neboli vizuální proměnné, zejména pozici, velikost, tvar, orientaci, barvu, případně průhlednost, texturu apod. Nejčastějšími druhy vizualizace dat jsou:

- graf
- diagram
- mapa
- grafický symbol (trojrozměrný symbol)

Estetické atributy grafů (vizuální proměnné):

- data
 - nezávislá a zaměnitelná část
 - přiřazení: k osám, tvaru, velikosti, barvě...
- transformace dat - výpočet hodnot pro
 - krabicový graf
 - hustotu pravděpodobnosti
 - agregovaná data
 - ...
- geometrický tvar
 - bod, čára, interval, boxplot...
- škála
 - určuje pro hodnoty dat, jak mají být zobrazena
 - obrácená funkce: čtení z grafu (pozice na osách, legendy)
- soustava souřadnic

- kartézská
- logaritmická
- polární
- mapové zobrazení

Příklady zobrazení jsou na následujících obrázcích.

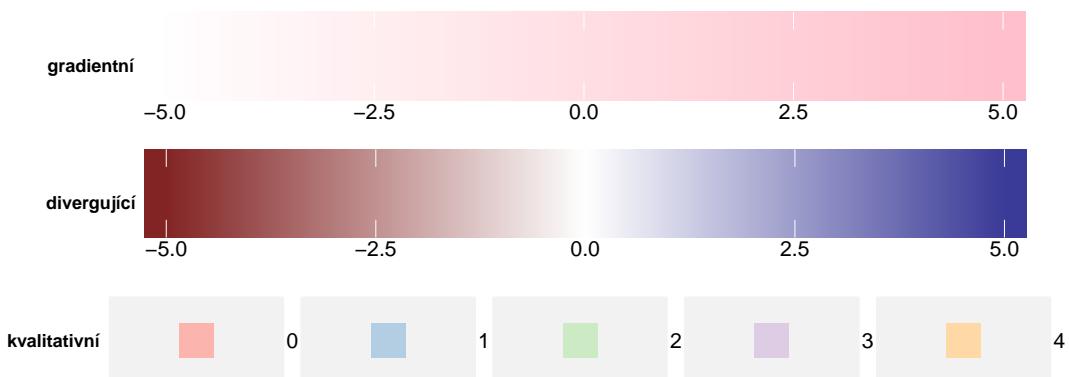
Velikost



Tvar



Barva



Průhlednost



Textura



Orientace



5.2.3 Typy základních grafů a jejich tvorba v R

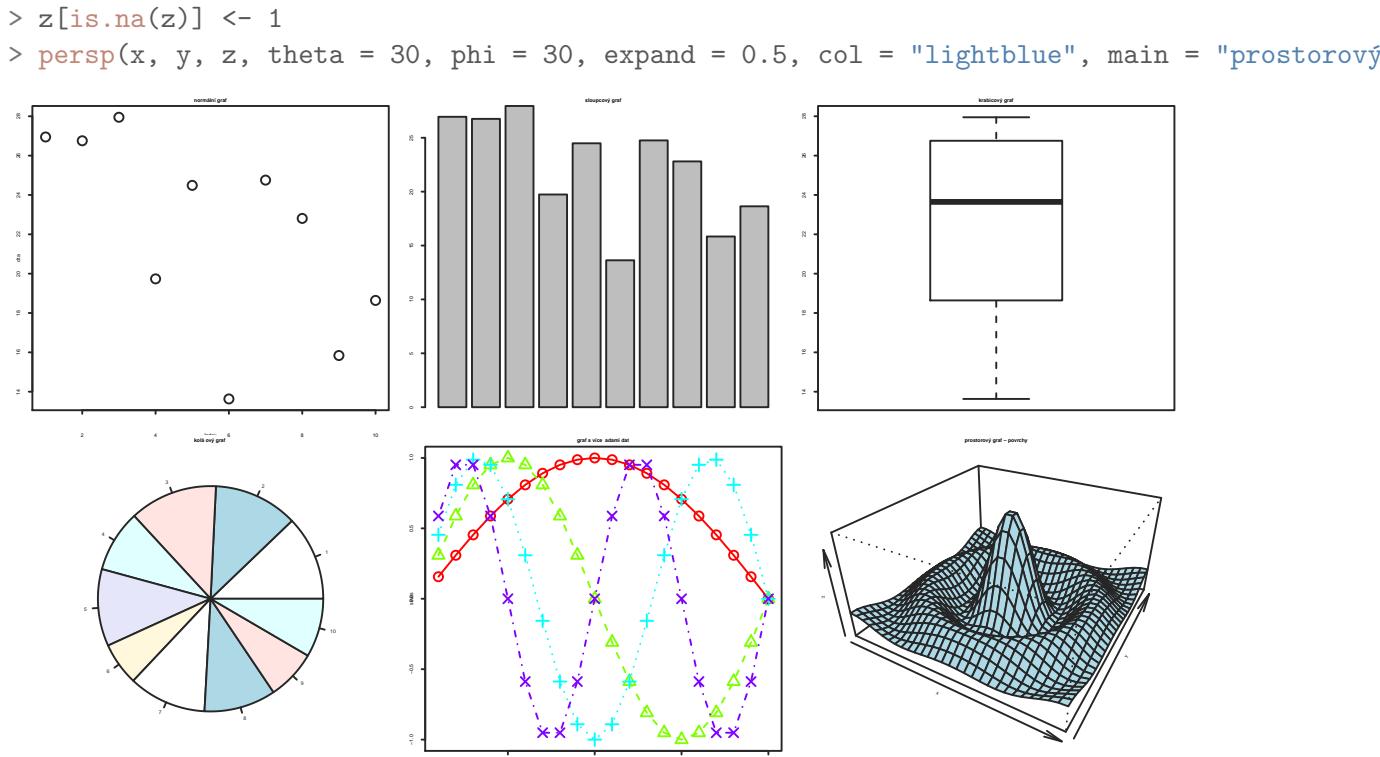
Vzhledem k velkému množství různých grafických prvků je dobré pro přehlednost uvést zjednodušený přehled základních grafických funkcí.

Základní typy grafů:

- plot (normální graf),
- barplot (sloupcový graf),
- boxplot (krabicový graf),
- pie (koláčový graf),
- histogram (sloupcový graf četnosti),
- matplot (graf s více řadami dat),
- persp (prostorový graf – povrchy).

ÚKOL 5.2 Vykreslete jednotlivé typy základních grafů

```
> dta = rnorm(10, mean = 20, sd = 5)
> plot(dta, main = "normální graf")
> barplot(dta, main = "sloupcový graf")
> boxplot(dta, main = "krabicový graf")
> pie(dta, main = "koláčový graf")
> sines <- outer(1:20, 1:4, function(x, y) sin(x/20 * pi * y))
> matplot(sines, pch = 1:4, type = "o", col = rainbow(ncol(sines)), main = "graf s více řadami dat")
> y <- x <- seq(-10, 10, length = 30)
> f <- function(x, y) {
+   r <- sqrt(x^2 + y^2)
+   10 * sin(r)/r
+ }
> z <- outer(x, y, f)
```



Obr. 5.1: Jednotlivé typy základních grafů

Dále je zde uveden přehled speciálních typů grafů:

pairs (skupiny XY grafů v jednom grafickém okně), stem (stonek s lístky), stars (hvězdový graf), dotchart (Clevelandův bodový graf), stripchart (pásový graf – 1D), sunflowerplot (slunečninový graf – body se shodnými souřadnicemi se vykreslí jako lístky vycházející z bodu), spineplot (speciální sloupcový graf s rozestupy a densitami), mosaicplot (mozaikový graf), fourfoldplot (čtyřlístkový graf), filled.contour (barevné kontury), contour (kontury), coplot (speciální matici XY grafů), cdplot (graf s výplní pod osou), bxp (jiný typ zadání boxplotu), assocplot (Cohen-Friendly graf), image (speciální typ grafu podobný filled.contour).

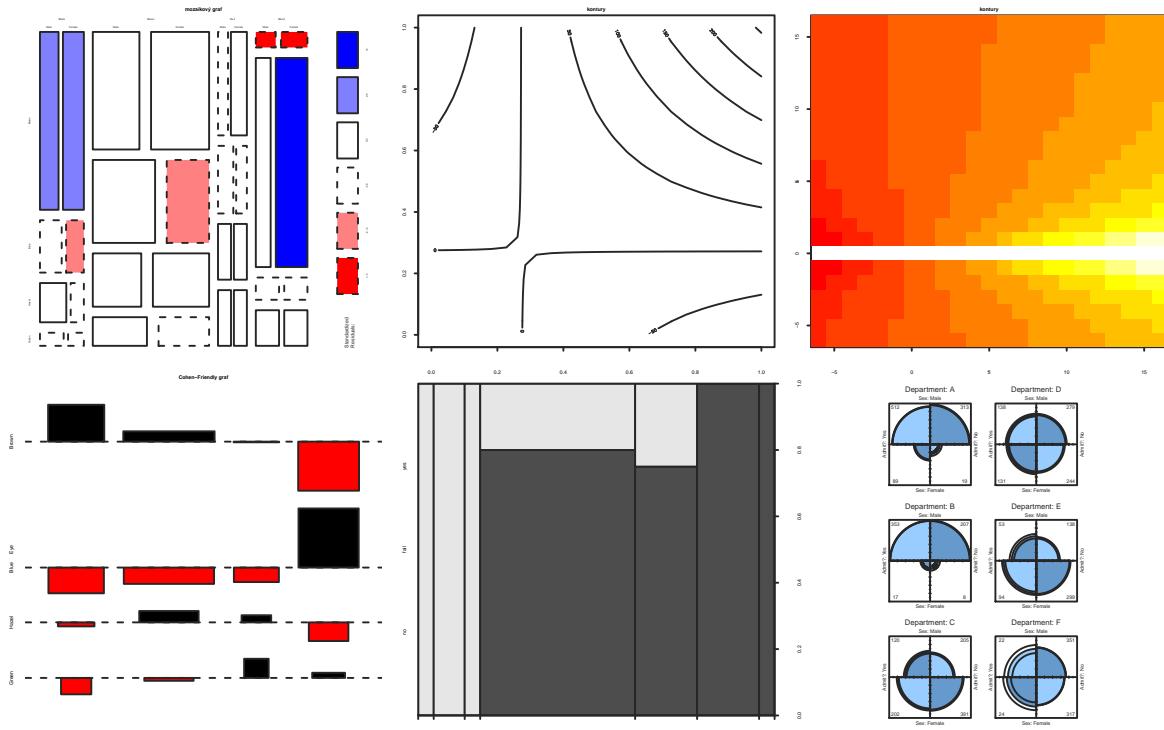
ÚKOL 5.3 Vykreslete alespoň 4 speciální typy grafů - použijte návod a data, která jsou volně dostupná

```
> dta = rnorm(10, mean = 20, sd = 5)
> mosaicplot(HairEyeColor, shade = TRUE, main = "mozaikový graf")
> ####
> x <- -6:16
> contour(outer(x, x), method = "edge", vfont = c("sans serif", "plain"), main = "kontury")
> z <- outer(x, sqrt(abs(x)), FUN = "/")
> image(x, x, z, main = "kontury")
> x <- margin.table(HairEyeColor, c(1, 2))
> assocplot(x, main = "Cohen-Friendly graf")
> ##
> fail <- factor(c(2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1,
+ 1, 1, 1), levels = c(1, 2), labels = c("no", "yes"))
```

```

> temperature <- c(53, 57, 58, 63, 66, 67, 67, 67, 68, 69, 70, 70, 70, 70, 72,
+   73, 75, 75, 76, 76, 78, 79, 81)
> spineplot(fail ~ temperature, main = "")
> ##
> x <- aperm(UCBAdmissions, c(2, 1, 3))
> dimnames(x)[[2]] <- c("Yes", "No")
> names(dimnames(x)) <- c("Sex", "Admit?", "Department")
> fourfoldplot(x, margin = 2)

```



Obr. 5.2: Příklad zobrazení speciálních typů grafů

V grafech je možné určité prvky vykreslit samostatně, a to především:

axis (osy), grid (mřížka), legend (legenda), rug (kartáč – vykresluje hustotu bodů), title (titulky a popisky), text (textová pole), points (body), lines (spojené čáry), segments (úsečky), abline (přímky), mtext (text na okraji grafu), matpoints (body ve více samostatných řadách), matlines (spojnice ve více samostatných řadách), curve (křivky, matematické funkce), box (obrys kolem grafu).

ÚKOL 5.4 Přidejte do základního grafu jednotlivé grafické prvky

```

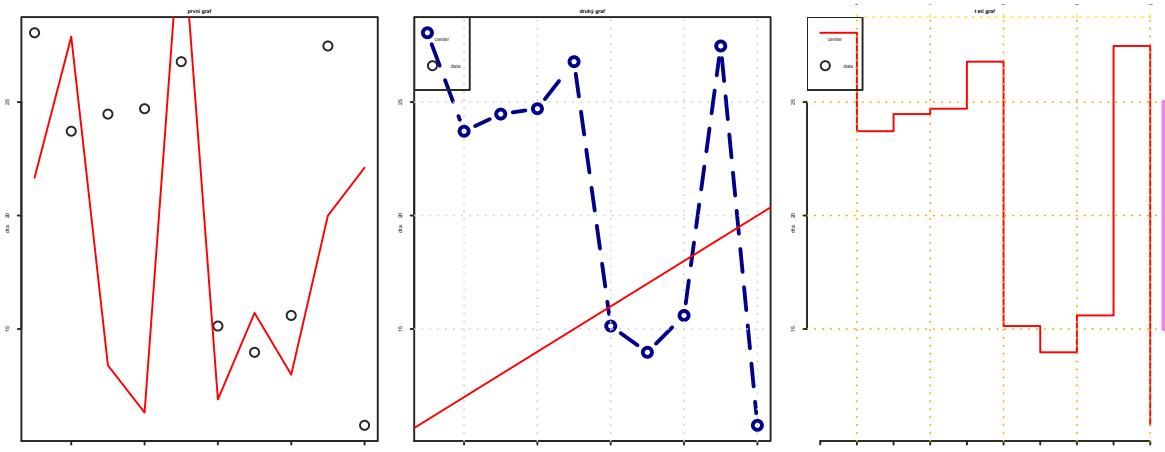
> dta = rnorm(10, mean = 20, sd = 5)
> ### 1.graf
> plot(dta, main = "první graf")
> lines(rnorm(10, mean = 20, sd = 5), col = "red")
> ### 2.graf
> plot(dta, type = "b", lty = 5, lwd = 2, col = "dark blue", main = "druhý graf")
> grid()

```

```

> legend("topleft", "data", pch = 1, title = "center")
> abline(10, 1, col = "red")
> ### 3.graf
> plot(dta, main = "třetí graf", type = "s", xaxt = "n", frame = FALSE, col = "red")
> axis(1, 1:10, LETTERS[1:10], col.axis = "blue")
> axis(4, col = "violet", col.axis = "dark violet", lwd = 2)
> axis(3, col = "gold", lty = 2, lwd = 0.5)
> grid(col = "orange")
> legend("topleft", "data", pch = 1, title = "center")

```



Obr. 5.3: Příklad zobrazení grafických doplňků

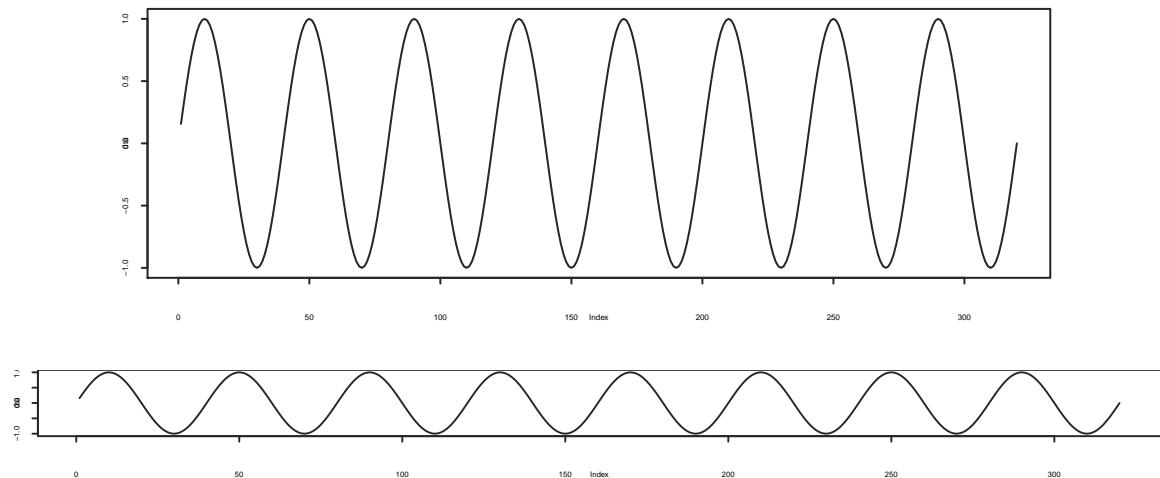
Nastavení grafiky R: windows (otevření a nastavení grafického okna), dev.set (výběr okna pro výstup), plot.window (nastavení koordinát, druhá, popř. třetí osa), par (nastavení parametrů grafického výstupu), split.screen (rozdělí okno, další close.screen, erase.screen), screen (obdélník), strwidth (počítá velikost textu v grafickém okně), locator (čte pozici kurzoru v grafickém okně), identify (identifikuje nejbližší vykreslený bod od pozice kurzoru), layout (nastaví rozdělení okna, víc nastavení než split.screen), frame (podobné plot.new, vytvoří grafické okno), xy.coords (souřadnice x a y), rgb (namíchá barvu), colors (přednastavené barvy), palette, rainbow, hcl, terrain.colors (palety barev), recordPlot (uložení grafu jako proměnné), plotmath (vykreslení matem. značek), windowsFons, Hershey (typy fontů), další funkce (balík grDevices).

5.2.4 Tvorba základních grafů

Při tvorbě high-level grafů je nutné dbát na několik důležitých věcí. High-level grafy většinou mohou jako data používat několik různých objektů a podle toho se potom chovají. Například funkce `plot` nejprve rozliší typ vstupujícího objektu a následně volá jinou funkci dle příslušného objektu (např. `plot.default` – základní graf, `plot.lm` – graf pro lineární model atd.). Velkou část parametrů, které nelze nastavit přímo jako argumenty dané funkce lze pak nastavit jako tzv. parametry grafického výstupu pomocí funkce `par`. Při přepnutí do okna pro grafiku se mění menu a kontextové menu tak, že umožňuje zkopírovat nebo uložit výslednou grafiku do schránky nebo do souboru v různých výstupních formátech. Údaje v palcích lze do cm převést funkcí `cm(x)`.

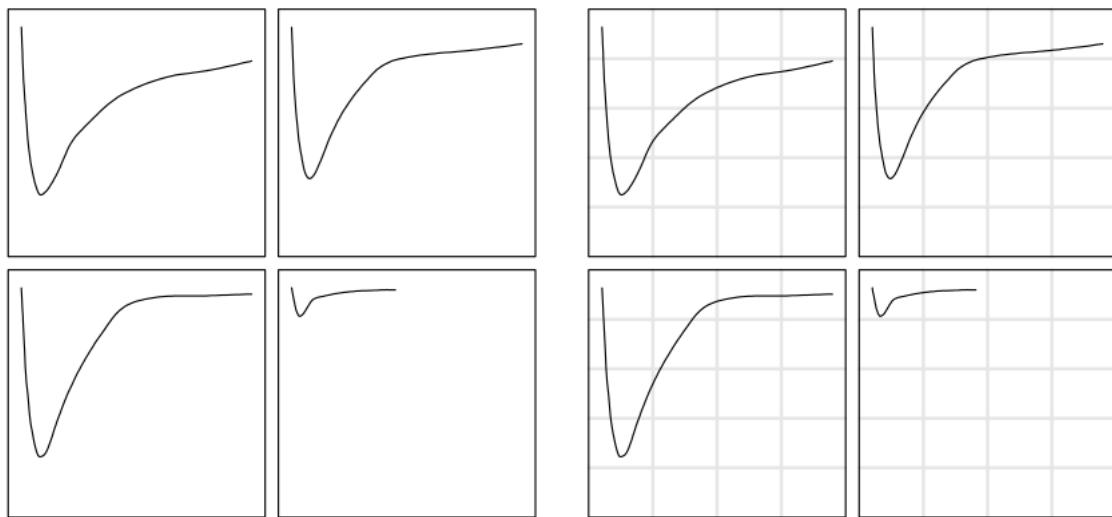
5.2.5 Způsob zobrazení a zavádějící grafy

Jak je důležitá forma prezentace výsledků, je vyjádřeno na následujících obrázcích, kdy na prvním z nich je zobrazen vliv poměru stran. Na obou obrázcích jsou vykreslena stejná data, avšak poměr stran se liší.



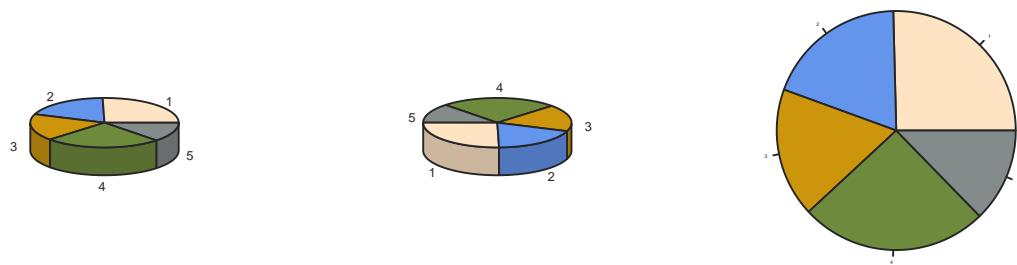
Obr. 5.4: Nevhodně zvolený poměr stran

Pro srovnání určitých hodnot v grafech je v některých případech vhodné do grafu zanést mřížku či vodící linky. Grafy mohou vypadat podobně, avšak data, která znázorňuje, mohou být velice odlišná.



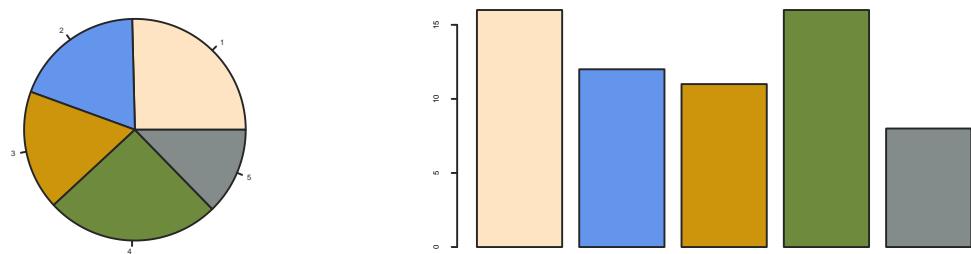
Obr. 5.5: Důležitost mřížky v grafech, plní srovnávací funkci

Pro prezentaci výsledků se často využívají grafy typu (pie - koláčový graf), který lze zobrazit prostorově. U tohoto zobrazení je potřeba dát si velký pozor na případ, kdy určitá hodnota je svojí pozicí v 3D grafu více zdůrazněna.



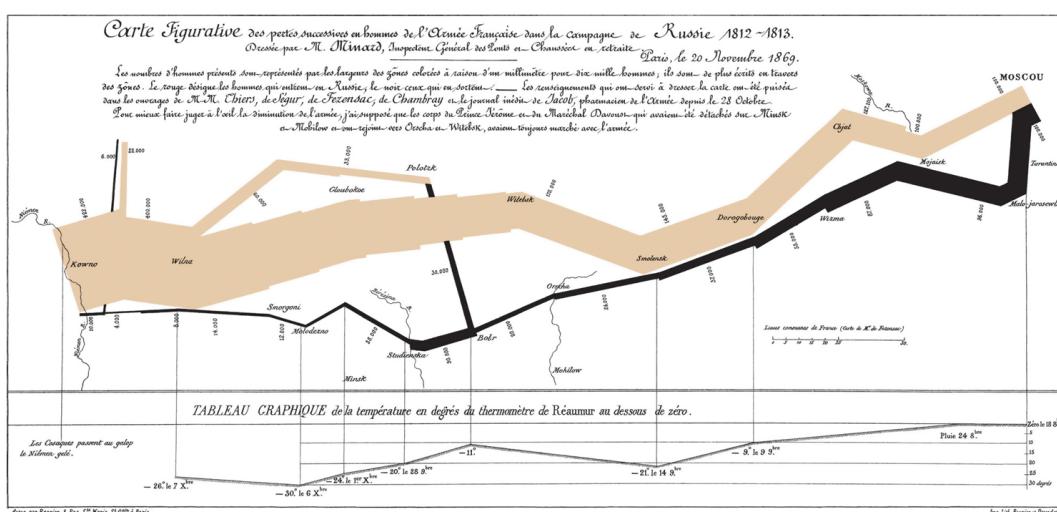
Obr. 5.6: Příklady nepřehlednosti koláčových grafů

Pro přehlednost je v někdy lepší zvolit graf typu barplot.

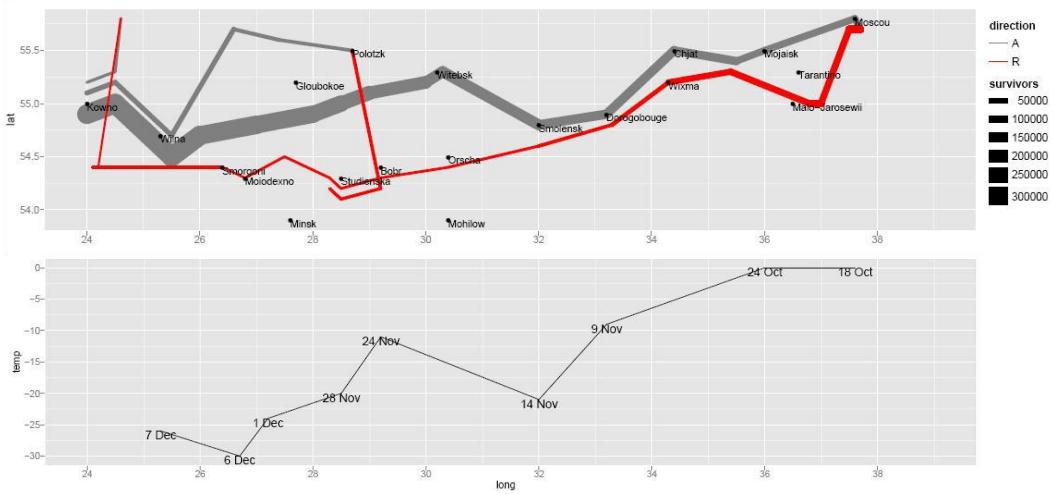


Obr. 5.7: Rozdíl mezi koláčovým a sloupcovým grafem

Na následujícím obrázku 5.8 je zobrazen postup Napoleonských vojsk na Moskvu a zpět. Je na něm vykreslen počet vojáků v podobě tloušťky čáry. Na spodním grafu je zobrazena teplota vzduchu. Graf poprvé uveřejnil Charles Joseph Minard, který je brán za ikonu informativních vizualizací v 19. století. Tento graf je považován za standard grafické prezentace, tzn. mnoho srozumitelných informací v jediném grafu, aniž by byl přeplněný a zmatečný.



Obr. 5.8: Minardova kresba postupu Napoleonských vojsk v letech 1812-13, zdroj:<http://en.wikipedia.org/wiki/File:Minard.png>



Obr. 5.9: Stejná kresba pomocí balíku ggplot2

5.2.6 R a GIS

V prostředí R je možno jednoduše zpracovávat data pro geografické informační systémy (GIS). Existuje velké množství R balíků, které jednotlivé operace podporují, mezi nejznámější patří balíky maptools, rgdal, rgeos, sp atd.

ÚKOL 5.5 Vykreslete rastrová data, která obsahují prostorové informace. Například zobrazte digitální model terénu, CORINE 2000 a sklonitost

6 Zpracování časových řad

6.1 Úvod do časových řad

Časová řada je posloupnost hodnot určitého statistického znaku (ukazatele) uspořádaná z hlediska času ve směru od minulosti k přítomnosti. Ukazatel musí být *věcně* a *prostorově* shodně vymezen po celé sledované období.

Obvykle prvním úkolem při analýze časových řad je získat rychlou a orientační představu o charakteru procesu, který tato řada reprezentuje. Mezi základní metody proto zcela běžně patří vizuální analýza chování ukazatele využívající grafů spolu s určováním elementárních statistických charakteristik. Pomocí vizuálního rozboru průběhu časových řad můžeme rozpoznat např. dlouhodobou tendenci v průběhu řady či některé periodicky se opakující vývojové změny apod. Tato analýza však nikdy nestačí k poznání hlubších souvislostí a mechanismů studovaného procesu a neumožňuje přehledným a koncentrovaným způsobem popsat jeho vlastnosti.

Časové řady dělíme na:

1 *Časové řady okamžikové* - hodnota ukazatele se plynule mění v čase, časová řada udává stav ukazatele v určitých okamžicích. Hodnoty stavu nemusí přímo záviset na délkách intervalu mezi odečty (při delším intervalu však může pochopitelně dojít k větší změně). Sčítání hodnot ukazatele této řady nemá logiku. Zde jsou příklady okamžikových časových řad:

- řada hodnot koncentrací nečistoty v odpadních vodách měřená v určitých intervalech na výstupu ze závodu
- řada teplot ovzduší na (hydro)meteorologické stanici odečítaná každou hodinu
- řada udávající počet zaměstnanců podniku na konci měsíců

2 *Časové řady intervalové* - hodnoty ukazatele sledují vznik nebo zánik (udávají přírůstek, úbytek) za časový interval, hodnoty tedy závisejí na délkách intervalů. Hodnotu ukazatele za delší interval lze získat sčítáním hodnot za dílčí části tohoto intervalu. Sčítání hodnot má logiku. Při vzájemném srovnávání hodnot ukazatele intervalové řady nutné konstantní délky intervalů (např. rok, čtvrtletí, měsíc a týden). Příklady intervalových časových řad:

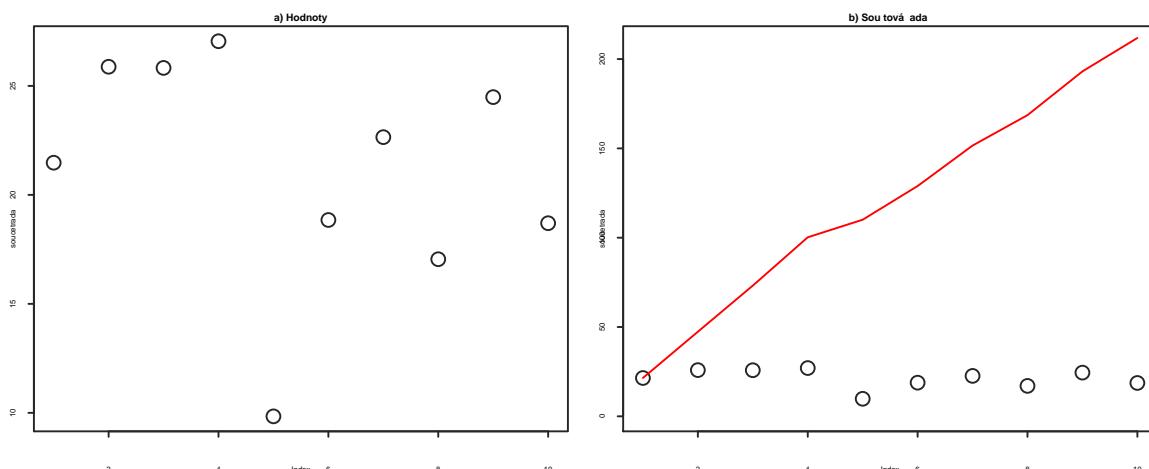
- postupná řada počtu narozených dětí ve státě za rok
- postupná řada hodnot měsíční průtoků

Odvozené časové řady existují pouze pro intervalové časové řady. Pro každou intervalovou časovou řadu lze z tzv. běžných hodnot sestrojit dva typy odvozených řad, a to:

- Součtová (kumulativní) řada
- Klouzavá řada

Na následujícím obrázku je zobrazen příklad zobrazení hodnot v grafu a součtová řada z těchto hodnot.

```
> plot(soucetrada, main = "a) Hodnoty")
> plot(soucetrada, ylim = c(0, 210), main = "b) Součtová řada")
> lines(cumsum(soucetrada), col = "red")
```



Obr. 6.1: Hodnoty a součtová řada - červeně

ÚKOL 6.1 Vypočítejte odvozenou klouzavou řadu. Tuto řadu společně z původními daty vykreslete do grafu a vypočítejte jejich průměry a směrodatné odchylinky.

6.2 Dekompozice časové řady

Dekompozice časové řady je metoda rozkladu časové řady na jednotlivé složky a je klasickou metodou analýzy časových řad. Spočívá v oddělení a analýze jednotlivých složek časové řady. Vývoj časové řady lze interpretovat jako výslednici (součet, event. součin) několika různých pohybů v čase. Část těchto pohybů má systematický (vypočítatelný) charakter, část pak nepravidelný charakter.

Složky časové řady y_t jsou:

- 1 Systematická složka Y_t
 - trendová složka - dlouhodobý základní směr vývoje řady T_t
 - krátkodobá
 - střednědobá - sezónní složka S_t
 - dlouhodobá - cyklická
- 2 Nepravidelná složka
 - skutečná - náhodná složka ϵ_t
 - odhadnutá - reziduální složka e_t

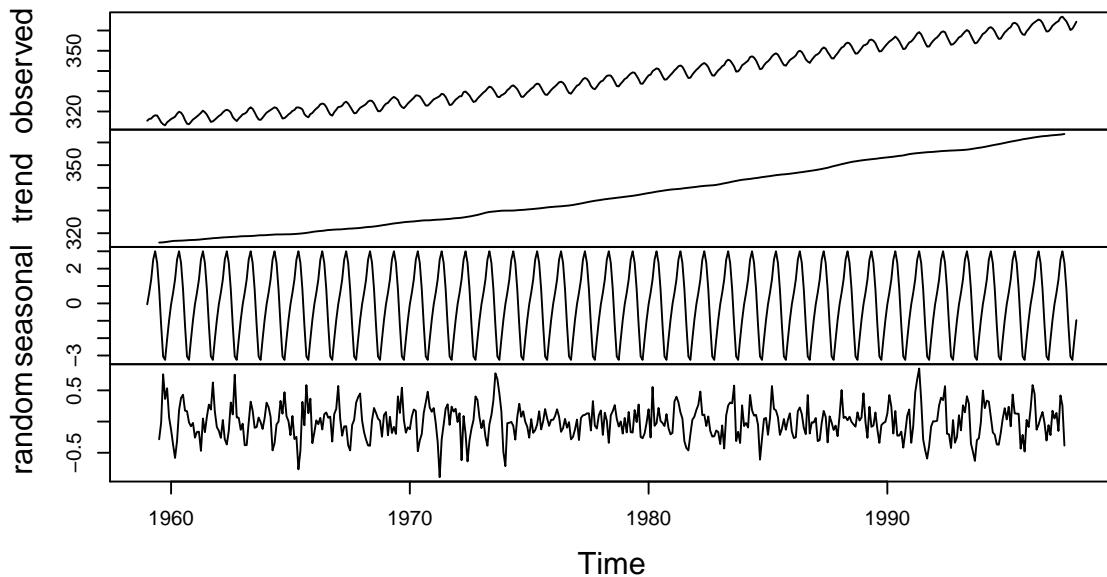
$$y_t = Y_t + \epsilon_t = T_t + S_t + \epsilon_t \quad (6.1)$$

Pro dekompozici časové řady se v prostředí R používá funkce `decompose`. Tato funkce rozdělí původní řadu na jednotlivé složky, a to na: *trendovou složku*, *sezónní složku* a *náhodnou složku*. Jejich charakter je možné vidět na obrázku 2.

ÚKOL 6.2 Proveďte dekompozici časové řady pomocí funkce `decompose` a podívejte se, jakou mají data strukturu, pomocí funkce `str`.

```
> m = decompose(co2)
> plot(m)
```

Decomposition of additive time series



```
> str(m)
List of 6
$ x      : Time-Series [1:468] from 1959 to 1998: 315 316 316 318 318 ...
$ seasonal: Time-Series [1:468] from 1959 to 1998: -0.0536 0.6106 1.3756 2.5168 3.0003 ...
$ trend   : Time-Series [1:468] from 1959 to 1998: NA NA NA NA NA ...
$ random  : Time-Series [1:468] from 1959 to 1998: NA NA NA NA NA ...
$ figure  : num [1:12] -0.0536 0.6106 1.3756 2.5168 3.0003 ...
$ type    : chr "additive"
- attr(*, "class")= chr "decomposed.ts"
```

Obr. 6.2: Ukázka dekompozice časové řady pomocí funkce decompose

V datech můžete vidět rozdělení původní časové řady na jednotlivé složky (trendová, náhodná a sezónní).

6.2.1 Aproximace trendu matematickými funkcemi

Typ nejvhodnější matematické funkce pro danou časovou řadu se určuje na základě předběžné analýzy řady, nejčastěji pomocí grafického záznamu řady nebo teoretických znalostí o průběhu trendové složky. Podle směru jej dělíme na trend:

- 1 rostoucí,
- 2 klesající,
- 3 střídavý - období růstu se střídají s obdobími poklesu,
- 4 časové řady, které trend postrádají.

Podle tvaru trendové složky jej dále dělíme na:

- 1 lineární - přímočarý,

2 nelineární - křivočarý.

Trendy mohou reprezentovat růst či pokles neomezený nebo omezený nějakou nepřekročitelnou konstantou (např. poptávka po zboží nemůže klesnout do záporných hodnot).

Systematická typologie funkcí vhodných pro popis trendové složky je uvedena např. v monografii J. Kozáka. Přehled funkcí:

- 1 Konstatní funkce - předpokládejme, že trendová složka je konstantní funkce:

$$T_t = \beta_0, \quad t = 1, 2, \dots, n. \quad (6.2)$$

Poté dostaneme bodový odhad $\hat{\beta}_0$ parametru β_0 pomocí metody nejmenších čtverců (viz. ??)

- 2 Lineární funkce - platí pro lineární trend:

$$T_t = \beta_0 + \beta_1 t, \quad t = 1, 2, \dots, n. \quad (6.3)$$

Bodové odhady $\hat{\beta}_0, \hat{\beta}_1$ parametrů β_0 a β_1 se také vyřeší pomocí metody nejmenších čtverců, kdy dostaneme soustavu dvou normálních rovnic.

- 3 Kvadratická funkce - v případě kvadratické funkce lze trend zapsat takto:

$$T_t = \beta_0 + \beta_1 t + \beta_2 t^2, \quad t = 1, 2, \dots, n. \quad (6.4)$$

Pro bodové odhady $\hat{\beta}_0, \hat{\beta}_1$ a $\hat{\beta}_2$ parametrů β_0, β_1 a β_2 dostaneme soustavu tří normálních rovnic.

- 4 Splinové funkce - v případě, kdy bylo potřeba popsat trend nějaké časové řady polynomem neúměrně vysokého stupně, rozdělíme časovou řadu na několik úseků a v každém z nich approximujeme trend polynomem nízkého stupně (např. prvního nebo druhého). Výsledná funkce je pak dáná spojením funkcí z jednotlivých úseků. Přitom požadujeme, aby tato funkce byla spojitá a navíc dostatečně hladká (má spojité derivace až do určitého rádu včetně).
- 5 a další, jako např. exponenciální, logistická, Gompertzova funkce, modifikovaná exponenciální funkce,..

6.2.2 Sezónní složka

Při analýze časové řady je třeba rozhodnout, zda má řada multiplikativní nebo aditivní sezónní složkou. Časová řada vykazuje multiplikativní sezónní složku, jestliže je amplituda sezónních fluktuací přímo úměrná úrovni trendu. Je-li však amplituda sezónních výkyvů prakticky nezávislá na úrovni trendu, je vhodné pracovat s aditivní sezónní složkou.

Hodnoty sezónní složky S se nazývají *sezónní faktory*. Jejich počet je dán počtem období (sezón) L v roce ($L = 4$ pro kvartální pozorování, $L = 12$ pro měsíční pozorování), označují se

$$S_{1+L_j}, S_{2+L_j}, \dots, S_{L+L_j}, \quad (6.5)$$

kde $j = 0, 1, \dots$ odpovídá postupně prvnímu, druhému, ... roku sledování časové řady. Hodnoty těchto faktorů se pro jednotlivé roky nemění. Pro jednoznačnost dekompozičního rozkladu se zpravidla požaduje, aby se vliv sezónních faktorů v rámci každého roku celkově vykompenzoval, proto se tyto faktory normalizují (normalizace sezónních faktorů). Multiplikativní sezónní faktory jsou bezrozměrná čísla. Pro jejich normalizaci se používají podmínky:

$$\sum_{i=1}^L S_{Lj} = L \quad \text{pro všechna} \quad j = 0, 1, \dots \quad (6.6)$$

Aditivní sezónní faktory se udávají ve stejných jednotkách jako hodnoty časové řady. Normalizační podmínka má tvar:

$$\sum_{i=1}^L S_{Lj} = 0. \quad (6.7)$$

Periodické kolísání představuje pravidelně se opakující výkyvy hodnot zkoumaného znaku střídavě oběma směry od hlavního vývojového směru. Je charakteristické délkou periody, velikostí výkyvu (amplitudou) a fázovým posunem (určuje polohu maxim a minim vzhledem k souřadnicím časové osy). Podle přítomnosti či nepřítomnosti periodické složky dělíme časové řady na:

- 1 periodické,
- 2 neperiodické.

Podle délky periody (frekvence) periodické složky se ekonomické časové řady rozlišují na krátkodobé, střednědobé neboli sezónní s periodou odpovídající přesně jednomu roku a dlouhodobé neboli cyklické s periodou delší než 1 rok. Největší význam vykazuje u velkého počtu časových řad právě *sezónní složka*.

Protože trendovou a sezónní složku lze většinou matematicky snadno popsat, shrnují se obvykle obě tyto složky pod společný pojem *systematická složka časové řady*.

Posouzení systematické složky se nazývá vyrovnávání (vyhlazování) časových řad. Nepravidelné kolísání je reprezentované náhodnou složkou časové řady, která je výslednicí různých nesledovaných nebo nepostřehnutelných vlivů.

Typy časových řad:

- 1 Stacionární řada — řada postrádající trendovou složku.
- 2 Nestacionární řada — řada s trendovou složkou.
- 3 Periodická řada — řada obsahující periodickou složku.
- 4 Neperiodická řada — řada neobsahující periodickou složku.

ÚKOL 6.3 Vypočítejte průměr periodické složky časové řady z předchozího úkolu.

6.3 Klouzavý průměr

Vyrovnávání časových řad se u této metody provádí po kratších klouzavých úsecích. Časovou řadu rozdělíme na klouzavé části o délce n . Úhrn příslušné klouzavé části (klouzavý úhrn) vydělíme délkom klouzavé části, čímž získáme *klouzavý průměr*. Vypočtený klouzavý průměr přiřadíme do středu klouzavé části:

- 1 při liché délce klouzavé části $n = 3, 5, 7, \dots$ nalezneme přímo prostřední období klouzavé části,
- 2 při sudé délce klouzavé části $n = 2, 4, 6, \dots$ padne klouzavý průměr mezi obě prostřední období.

Definitivní klouzavé průměry pro sudou délku klouzavé části vypočteme jako průměry dvou sousedních klouzavých průměrů — tuto operaci nazýváme centrování (rozlišujeme pak *necentrované* a *centrované* klouzavé průměry).

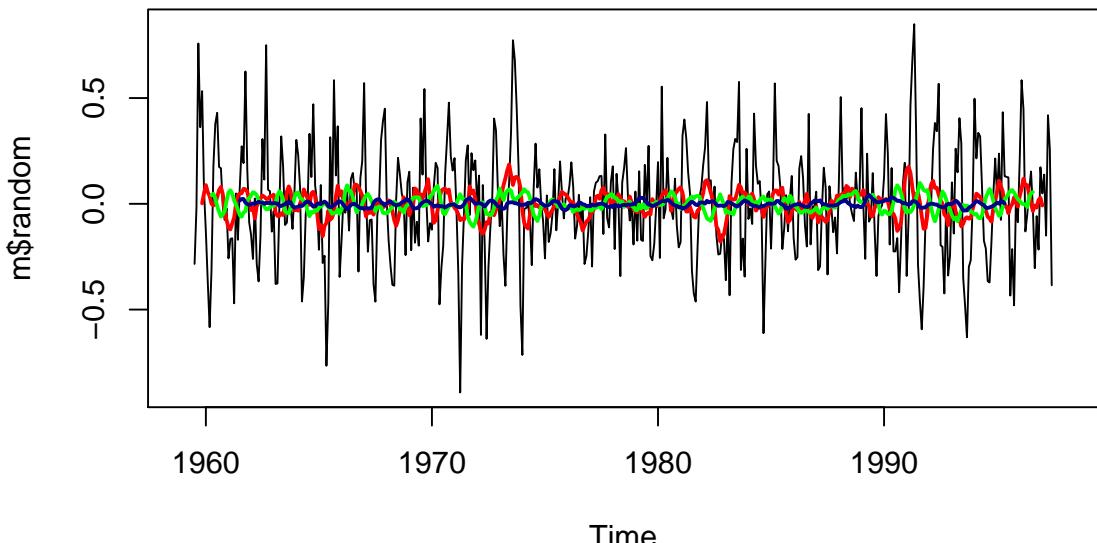
S rostoucí délkou klouzavé části:

- roste vyhlažující účinek klouzavých průměrů,
- prodlužuje se nevyrovnaná část na začátku a konci řady.

Klouzavý průměr v prostředí R lze vytvořit pomocí vlastní funkce, která využívá funkci `filter` nebo pomocí funkce `rollmean`, která je součástí balíku `zoo`.

ÚKOL 6.4 Vytvořte klouzavý průměr pro náhodnou složku dekomponované časové řady z příkladu 2. Filtr nastavte na hodnoty 10, 20 a 50.

```
> plot(m$random)
> ## vytvoření funkce pro klouzavý průměr
> ma <- function(x, n) {
+   filter(x, rep(1/n, n), sides = 2)
+ }
> lines(ma(m$random, 10), col = "red", lwd = 2)
> lines(ma(m$random, 20), col = "green", lwd = 2)
> lines(ma(m$random, 50), col = "dark blue", lwd = 2)
```



Obr. 6.3: Ukázka klouzavých průměrů s nastavením filtru na hodnoty 10, 20 a 50

6.4 Korelace a regrese

6.4.1 Kovariance a korelace

Číselně lze sílu lineární závislosti mezi dvěma kvantitativními znaky popsat pomocí *kovariance* a ve standardizované formě pomocí *korelace*. Výběrová kovariance mezi veličinami x a y se vypočte následovně:

$$s_{x,y} = cov(x,y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (6.8)$$

Teoreticky může kovariance nabývat všech reálných hodnot, kdy růst závislosti se projevuje v ros-

toucí hodnotě absolutní kovariance. Z tohoto důvodu je lepší pro posouzení lineární závislosti použít standardizovanou kovarianci neboli *korelační koeficient*, která se vypočte následovně:

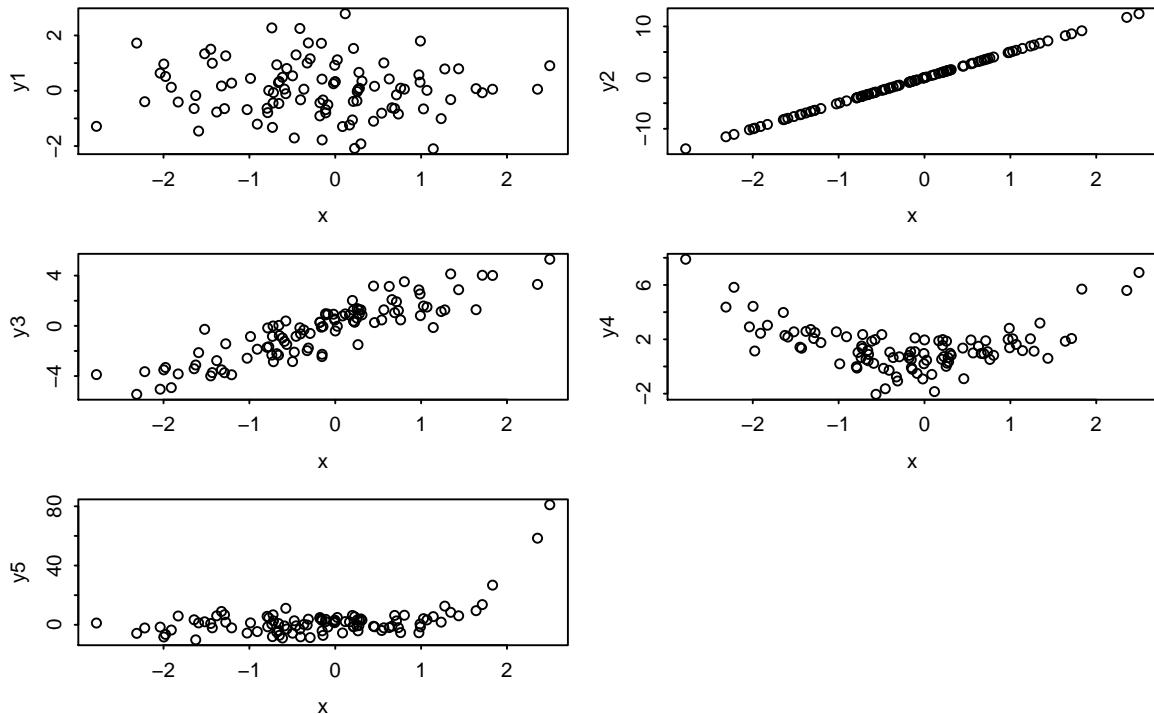
$$\text{cor}(x,y) = \frac{s_{x,y}}{s_x \cdot s_y}, \quad (6.9)$$

je to tedy podíl kovariance a směrodatných odchylek pozorovaných znaků.

Korelační koeficient může nabývat pouze hodnot od -1 do 1 a platí následující:

- $\text{cor}(x,y) = 1$, mezi x a y existuje přesná *rostoucí* lineární závislost,
- $\text{cor}(x,y) = -1$, mezi x a y existuje přesná *klesající* lineární závislost,
- $\text{cor}(x,y) = 0$, dá se říci, že mezi x a y neexistuje lineární závislost.

ÚKOL 6.5 Určete přibližně jaká je hodnota korelačního koeficientu následujících grafů.



ÚKOL 6.6 Vypočítejte hodnotu korelačního koeficientu pro x a y , kdy $x=rnorm(100, 13)$ a $y=(2/x) + rnorm(100, 50)$.

6.4.2 Autokorelace

Získat poznatek o struktuře sledovaného procesu lze získat pomocí autokorelační funkce (ACF), respektive pomocí paraciální autokorelační funkce (PACF). Autokorelační funkce ukazuje, jak korelace mezi dvěma libovolnými členy řady závisí na vzdálenosti těchto členů. V případě stacionárního stochastického procesu je autokorelační funkce definována vztahem:

$$\rho_k = \frac{\text{cov}(X_t, X_{t+k})}{D(X_t)}, \quad (6.10)$$

kde $D(X_t)$ je rozptyl dané časové řady.

V případě stacionárního stochastického procesu X_t má autokorelační funkce následující vlastnosti:

- $\rho_0 = 1$, což je autokorelační funkce 0-tého řádu je rovna jedné,
- $|\gamma_k| \leq \gamma_0 ; |\rho_k| \leq 1$, tzn. absolutní hodnota autokovarianční funkce $k - t$ ého řádu je menší nebo rovna hodnotě autokovarianční funkci 0-tého řádu a absolutní hodnota autokorelační funkce „ k -tého“ řádu je menší nebo rovna jedné,
- $\gamma_k = \gamma_{-k}$ a $\rho_k = \rho_{-k}$ pro všechna k a je tedy symetrická kolem $k=0$.

Graf autokorelační funkce se nazývá koreogram.

Korelace mezi dvěma náhodnými veličinami je často způsobena tím, že obě tyto veličiny jsou korelovány s veličinou třetí. Velká část korelace mezi veličinami X_t a X_{t-k} může být tedy způsobena jejich korelací s veličinami $X_{t-1}, X_{t-2}, \dots, X_{t-k+1}$. Parciální autokorelace podávají informaci o korelací veličin X_t a X_{t-k} očištěné o vliv veličin ležících mezi nimi. Parciální autokorelace poskytují parciální korelační koeficient, které podávají informaci o tzv. parciální autokorelační funkci se zpožděním k tzn. informaci o korelaci veličin X_t a X_{t-k} očištěné o vliv veličin ležících mezi těmito veličinami.

6.4.3 Regresní analýza

Smyslem regresní analýzy je zejména nalézt formu systematického funkčního vztahu mezi odezvou a jedním nebo více prediktory. Pokud se toto podaří, slouží získaný model k následujícím účelům:

- 1 funkční popis závislosti,
- 2 predikci,
- 3 nastavení standardů (kontrola procesů).

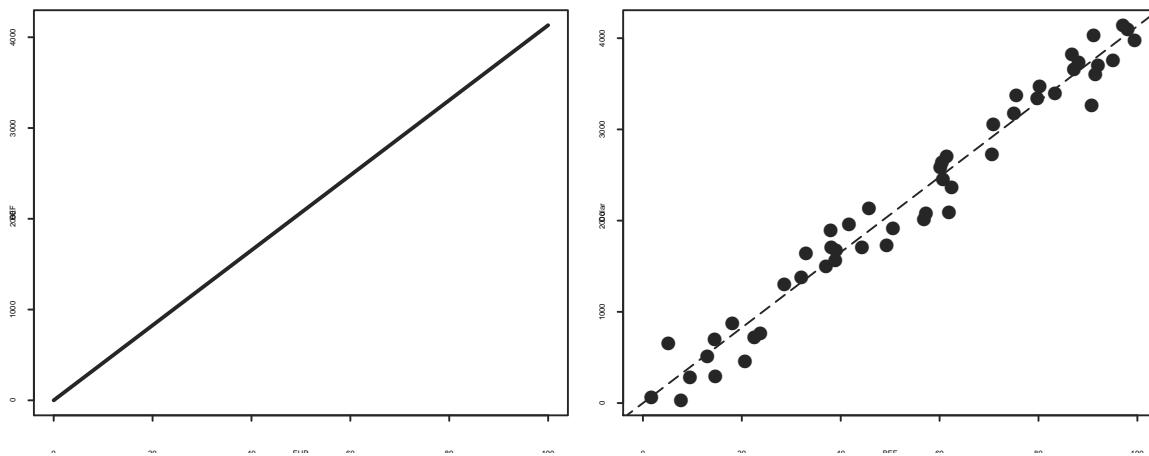
Regressa s jedním prediktorem

V přírodních vědách, matematice a fyzice lze často vztah mezi dvěma veličinami Y a z , pomocí funkčního vztahu jedinou reálnou funkcí:

$$Y = f(z). \quad (6.11)$$

ÚKOL 6.7 Vykreslete graf závislosti eura na belgickém franku.

```
> set.seed(1901)
> x <- runif(50, 0, 100)
> b <- 41.325
> sigma <- 200
> plot(grid, b * grid, type = "l", xlab = "EUR", ylab = "BEF", lwd = 2)
> plot(x, b * x + rnorm(length(x), 0, sigma), type = "p", pch = 16, xlab = "BEF",
+       ylab = "Dolar", xlim = c(0, 100))
> abline(0, b, lty = 5, lwd = 1)
```



Forma modelu

Nejjednodušší formou regresního modelu je *lineární regresní model*. Obdobně jako u approximace trendu (systematické složky) matematickými funkcemi (viz. kapitola ??) lze provádět regresní analýzu.

Regresní funkce

- Prokládá (vyrovnává) hodnoty přímkou
- Zkoumá účinek nezávisle proměnné x na závisle proměnnou y
- Regresní rovnice
 - Známe-li hodnoty x , pak můžeme odhadnout y

$$y = a + bx (+e) \quad (6.12)$$

- y ... předpokládaná hodnota y
- x ... pozorovaná hodnota x
- a ... průsečík (konstanta)
- b ... směrnice přímky
- a, b ... parametry regresní rovnice

Podmínky regresní analýzy

- 1 Normalita závisle proměnné
- 2 Normalita chyb
- 3 Nezávislost pozorování
 - Hodnota pozorování nezávisí na okolních případech
 - Problematické - časové řady, prostorová data
- 4 Homoskedasticita
 - Rozptyl náhodné složky je konstantní
 - Graf reziduů
- 5 Nekorelovanost proměnných
 - multikolinearita
- 6 + další (např. diagnostika, v praxi se na ně nebere ohled)

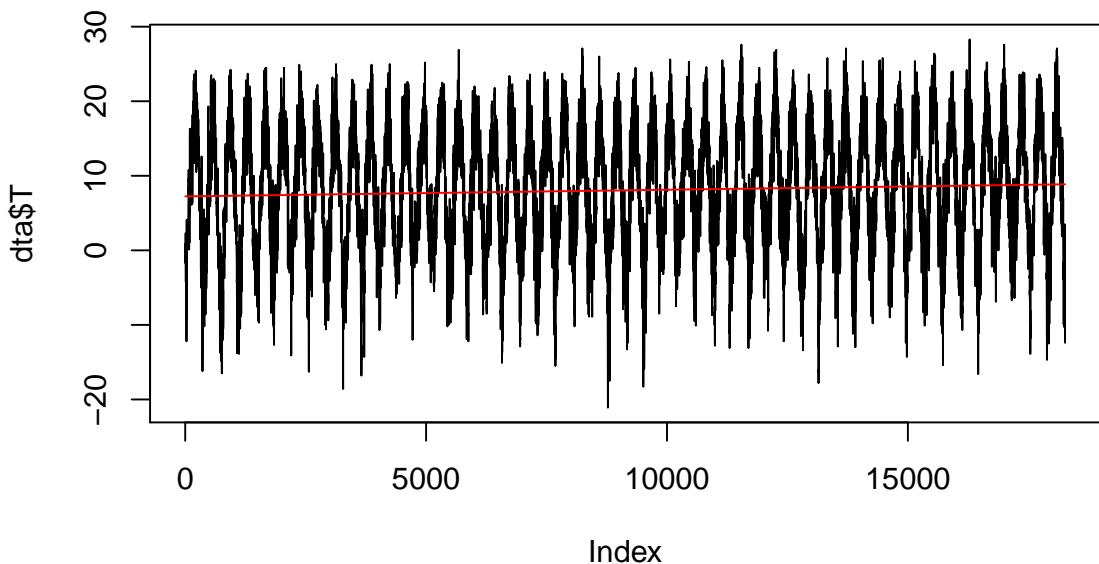
V prostředí R se lineární modely vytváří pomocí funkce `lm()`.

ÚKOL 6.8 Vytvořte lineární model pro teplotu vzduchu a času, pomocí funkce `str()` se podívejte, jak vypadá datová struktura vašeho lineárního modelu.

```
DTM PRECIP TEMP
1: 1961-01-01      0.0 -0.7
2: 1961-01-02      0.0 -1.7
3: 1961-01-03      0.2  0.2
4: 1961-01-04      0.0  0.8
5: 1961-01-05      0.1  0.9
6: 1961-01-06      0.0 -0.4

> lin_mod = lm(dta$T ~ dta$DTM)
> plot(dta$T, type = "l")
> lines(lin_mod$fitted.values, col = "red")

PackageWarningboxed is void – discard it. -
```



```
> lin_mod

Call:
lm(formula = dta$T ~ dta$DTM)

Coefficients:
(Intercept)      dta$DTM
7.533218        0.000089
```

6.5 Metoda nejmenších čtverců

Ve všech případech, kdy z hodnot měřené závislosti dvou fyzikálních veličin zatížených chybami určujeme její nejpravděpodobnější průběh, mluvíme o vyrovnání funkční závislosti. Toto vyrovnání lze provádět graficky i numericky. Nejnámější numerická vyrovnávací metoda je metoda nejmenších čtverců. Metoda nejmenších čtverců slouží k nalezení takového vyrovnání měření, aby součet druhých mocnin chyb nalezeného řešení byl minimální. Zjednodušeně, aby součet čtverců odchylek byl nejmenší.

Pro proložení měřených hodnot přímkou s rovnicí:

$$y = f(u) = k_1 u + k_0, \quad (6.13)$$

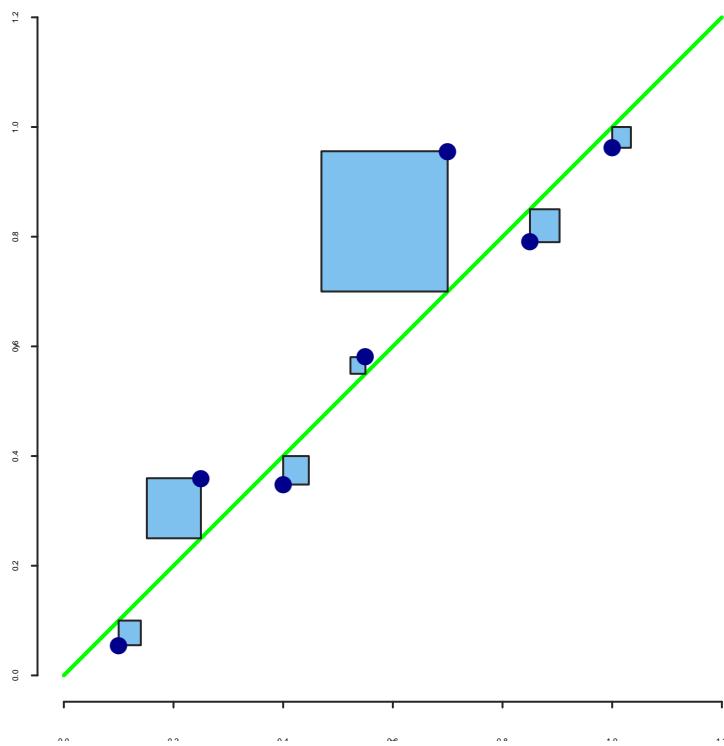
lze vypočítat koeficienty k_0 a k_1 dle následujících vzorců:

$$k_1 = \frac{n \sum u_i y_i - \sum u_i \sum y_i}{n \sum u_i^2 - (\sum u_i)^2}, \quad (6.14)$$

$$k_0 = \frac{\sum u_i^2 \sum y_i - \sum u_i \sum u_i y_i}{n \sum u_i^2 - (\sum u_i)^2} \quad (6.15)$$

ÚKOL 6.9 Vykreslete vzorový graf metody nejmenších čtverců.

```
> set.seed(733)
> sigma <- 0.1
> a <- 0
> b <- 1
> xx <- seq(0, 1.2, by = 0.01)
> yx <- a + b * xx
> x <- c(0.1, 0.25, 0.4, 0.55, 0.7, 0.85, 1)
> yhat <- a + b * x
> y <- yhat + rnorm(7, 0, sigma)
> eps <- abs(y - yhat) * 0.9
> par(mfrow = c(1, 1), bty = "n", mar = c(4, 4, 1, 1) + 0.1, pch = 16)
> par(par.plot)
> plot(xx, yx, type = "l", lty = 1, lwd = 2, xlab = "x", ylab = "y", xlim = c(0,
+ 1.2), col = "green")
> for (i in 1:length(x)) {
+   if (y[i] >= yhat[i]) {
+     rect(x[i] - eps[i], yhat[i], x[i], y[i], lwd = 1, col = "skyblue2")
+   } else {
+     rect(x[i], y[i], x[i] + eps[i], yhat[i], lwd = 1, col = "skyblue2")
+   }
+ }
> points(x, y, cex = 1.2, col = "darkblue")
```



Předpoklady metody nejmenších čtverců:

- 1 Regresní parametry mohou teoreticky nabývat jakýchkoli hodnot.
- 2 Regresní model je lineární v parametrech.
- 3 Jednotlivé nezávislé proměnné jsou skutečně vzájemně nezávislé, tedy mezi nimi nedochází k tzv. multikolinearitě.
- 4 Podmíněný rozptyl $D(y/x) = 2$ je konstantní (tzv. podmínka homoskedasticity).
- 5 Náhodné chyby mají nulovou střední hodnotu, mají konečný rozptyl a jsou nekorelované.

7 Odhad parametrů a testování hypotéz

Jedním ze základních úkolů statistiky ve vodním hospodářství je extrakce informací o hydrologických procesech na základě pozorování. Tedy získávání informací o náhodné veličině X na základě výběru (x_1, x_2, \dots, x_N) . K analýze výběru můžeme přistupovat v zásadě dvojím způsobem - buď může být cílem odhad nějakých parametrů veličiny X , nebo testování hypotéz o těchto parametrech. Podstatná část diskuze uvedené v této kapitole vychází z knihy *Statistická analýza v problematice klimatu* (Von Storch a Zwiers, 2001).

7.1 Odhad parametrů a jeho vlastnosti

V praxi se často snažíme získat z výběru (pozorování) odhad parametru popisující nějakou vlastnost náhodné veličiny. Tj. snažíme se odhadnout hodnotu tohoto parametru pomocí nějaké funkce výběru, označovanou *estimator*. Její hodnotou může být buď číslo (*point estimator*) nebo interval (*interval estimator*). Ideálně je bodový odhad v okolí skutečné hodnoty parametru a velikost tohoto okolí se zmenšuje s rostoucí velikostí výběru. Podobně intervalový odhad je vytvořen tak, aby při opakováném samplování pokrýval hodnotu skutečného parametru s pevně danou (zpravidla vysokou, např. 90%) pravděpodobností, tedy velikost intervalu se zmenšuje s velikostí výběru. Typicky odhadujeme momenty rozdělení, nicméně je možné odhadovat i celé rozdělení pravděpodobnosti nebo doby opakování a N -letá maxima.

Odhad neznámého parametru, např. α , označujeme dále jako $\hat{\alpha}$. Funkce, která odhad zprostředkovává, se zpravidla v češtině nazývá „odhad“, stejně jako konkrétní odhad pomocí této funkce. V anglicky psané literatuře se rozlišuje odhadující funkce *estimator* od konkrétního odhadu *estimate*. Aby toto rozlišení bylo v textu zřejmé, používáme dále pro odhadující funkci výraz „estimator“, který se v češtině volně používá též.

Příkladem estimátorů jsou charakteristiky výběru uvedené v kapitole 4 (např. empirické momenty, empirická distribuční funkce, histogram atd.). V této části se budeme obecně zabývat vlastnostmi odhadů, ne konkrétními odhady.

Spolehlivost odhadu

Jak bylo řečeno, jedním z požadavků na odhad je, aby ležel v okolí skutečné hodnoty. Velikost tohoto okolí lze definovat například pomocí střední kvadratické chyby:

$$\text{MSE}(\alpha, \hat{\alpha}) = E[(\alpha - \hat{\alpha})^2] \quad (7.1)$$

kde α je odhadovaný parametr a $\hat{\alpha}$ je odhad tohoto parametru, získaný libovolným způsobem. Pokud pro dva různé estimátory $\hat{\alpha}_1$ a $\hat{\alpha}_2$ a pro libovolnou hodnotu α platí, že $\text{MSE}(\alpha, \hat{\alpha}_1) < \text{MSE}(\alpha, \hat{\alpha}_2)$ říkáme, že estimátor $\hat{\alpha}_1$ je spolehlivější než estimátor $\hat{\alpha}_2$ a zároveň odhad $\hat{\alpha}_1$ je spolehlivější než odhad $\hat{\alpha}_2$.

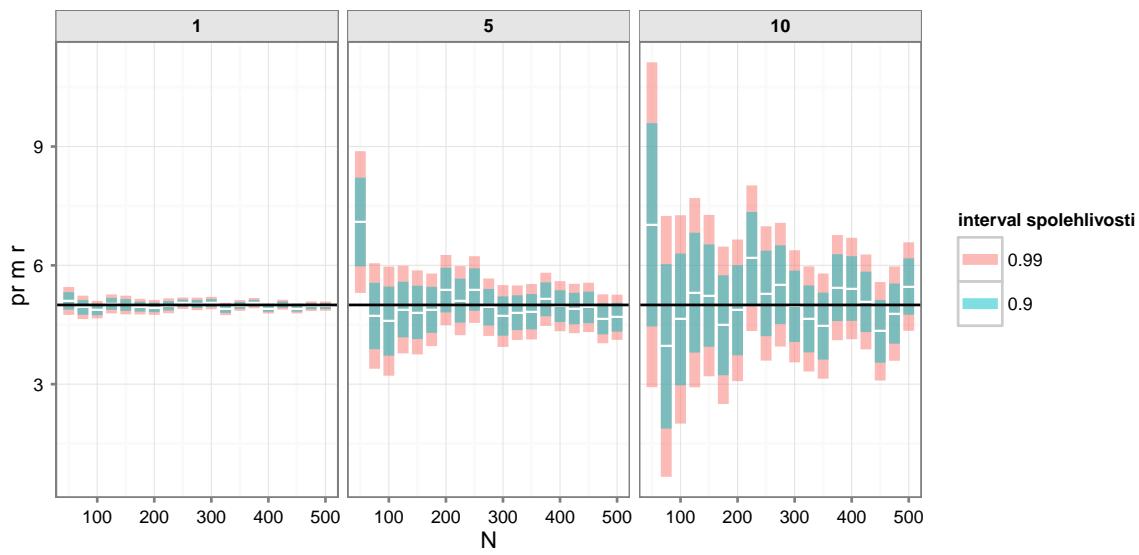
Pro některé parametry známe rozdělení pravděpodobnosti odhadů parametrů a můžeme tedy přímo vyčíslit interval spolehlivosti pokrývající skutečnou hodnotu neznámého parametru s předem definovanou pravděpodobností. Typickým příkladem je odhad intervalu spolehlivosti střední hodnoty normálního rozdělení na základě výběru:

$$\bar{x} \pm t_{\alpha/2}(N-1) \frac{s_x}{\sqrt{N}}, \quad (7.2)$$

kde \bar{x} a s_x je výběrový průměr a směrodatná odchylka, N je velikost výběru, a $t_{\alpha/2}(N-1)$ je $\alpha/2$ -tý kvantil Studentova t-rozdělení s $N-1$ stupni volnosti. Hledaný parametr pak leží v tomto intervalu se $100(1-\alpha)$ procentní pravděpodobností.

ÚKOL 7.1 Spočítejte 90% a 99% interval spolehlivosti odhadu střední hodnoty rozdělení veličiny X na základě výběru o velikosti N z $n1:n2 = 50:500$. Výběr generujte z normálního rozdělení se střední hodnotou 5 a směrodatnou odchylkou 1, 5 a 10.

```
> require(data.table)
> require(ggplot2)
> require(reshape2)
> n1 = 50
> n2 = 500
> ns = seq(n1, n2, by = 25)
> sd = c(1, 5, 10)
> inte = array(NA, dim = c(length(ns), length(sd), 2, 3),
+               dimnames = list(N = ns, s = sd, cl = c(.9, .99), interval = c('LO', 'MEAN', 'UP')))
> cl = c(.9, .99)
>
> for (N in ns){
+   for (s in sd){
+
+     x = rnorm(N, 5, s)
+     me = mean(x)
+
+     for (a in cl){
+       T = qt((1 - a) / 2, N - 1) * sd(x) / sqrt(N) # testovací statistika dle rovnice 5.2
+       inte[which(ns == N), which(sd == s), which(cl == a), ] = me + c(-T, 0, T)
+     }
+   }
+ }
>
> res = data.table(melt(inte))
> res = cbind(res[interval=='LO', list(N, s, cl, LO = value)],
+             res[interval=='MEAN', list(MEAN = value)], res[interval=='UP', list(UP = value)])
>
> ggplot(res) +
+   theme_plot + # není nutné - jen téma pro tisk
+   geom_linerange(aes(x = N, ymin = LO, ymax = UP,
+                      col=factor(cl, levels = c('0.99', '0.9'))), alpha = 0.5, lwd = 2) +
+   geom_point(aes(x = N, y = MEAN), col = 'white', pch = '-', size=4) +
+   facet_grid(~s) + geom_hline(yintercept = 5) +
+   scale_colour_discrete(name = 'interval spolehlivosti') +
+   ylab('průměr')
```



Obr. 7.1: Odhadý střední hodnoty výběru s intervaly spolehlivosti pro různé velikosti výběrů a různé směrodatné odchyly.

Jak vyplývá z rovnice 7.2, je interval spolehlivosti tím užší, čím větší je výběr, a zároveň tím širší, čím větší je rozptyl. To potvrzuje i obrázek.

□

ÚKOL 7.2 Vykreslete 90% interval spolehlivosti pro odhad střední hodnoty veličiny z Normálního rozdělení se střední hodnotou 0 a rozptylem 1 pro velikosti výběrů 10, 20, ..., 100. Pro každou velikost výběru simulujte 100krát data ze zadанého rozdělení a spočítejte výběrový průměr, tj. pro každou velikost výběru budeme mít 100 průměrů. Můžeme předpokládat, že všechny tyto průměry leží uvnitř intervalů spolehlivosti? Kolik hodnot (ze sta průměrů) očekáváme, že bude ležet mimo intervaly spolehlivosti a proč? Bude očekávání jiné v případě jiného intervalu spolehlivosti?

```
> require(data.table)
> require(ggplot2)
> require(reshape2)
>
> cl = .9
> ns = seq(10, 100, by = 10)
> T = qt((1-cl)/2, ns-1) * 1/sqrt(ns)
> rngs = data.table(N = ns, LO = -abs(T), UP = abs(T))
> est = data.table()
>
> for (n in ns){
+   sampl = replicate(100, mean(rnorm(n, 0, 1)))
+   out = sampl > rngs[N == n, UP] | sampl < rngs[N == n, LO]
+   est = rbind(est, data.table(N = n, EST = sampl, OUT = out))
+ }
>
> rngs[, CNT:=est[, length(which(OUT == TRUE)), by = N][, V1]]
```

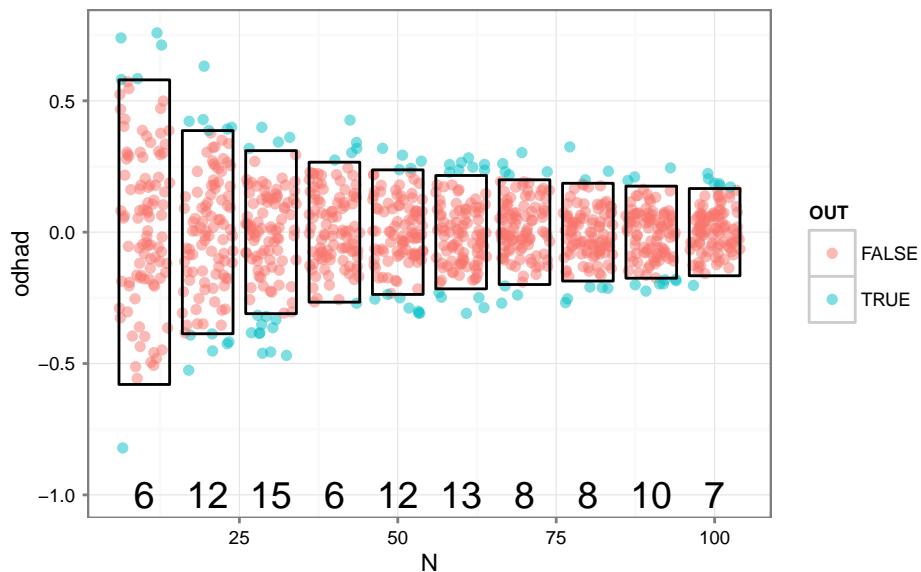
N	LO	UP	CNT
1: 10	-0.5797	0.5797	6
2: 20	-0.3866	0.3866	12
3: 30	-0.3102	0.3102	15
4: 40	-0.2664	0.2664	6
5: 50	-0.2371	0.2371	12
6: 60	-0.2157	0.2157	13

```

7: 70 -0.1993 0.1993 8
8: 80 -0.1861 0.1861 8
9: 90 -0.1752 0.1752 10
10: 100 -0.1660 0.1660 7

> ggplot(est) +
+   theme_plot +
+   geom_jitter(aes(x = N, y = EST, col = OUT), alpha = 0.5) +
+   geom_text(aes(x = N, y = -1, label = CNT), data = rngs) +
+   ylab('odhad') +
+   geom_rect(aes(xmin = N - 4, xmax = N + 4, ymin = LO, ymax = UP),
+             col='black', data = rngs, fill = NA)

```



Obr. 7.2: Intervaly spolehlivosti pro veličinu z $N(0, 1)$ a různé velikosti výběru. Tečky odpovídají vypočteným průměrům ze 100 samplů generovaných pro jednotlivé velikosti výběru.

□

Nestrannost

Systematická chyba odhadu B odhadu $\hat{\alpha}$ parametru α je definována jako

$$B(\hat{\alpha}) = E(\hat{\alpha}) - \alpha \quad (7.3)$$

Nestranný odhad je takový, pro nějž je systematická chyba nulová, tj. $E(\hat{\alpha}) = \alpha$. Pozitivně vychýlený odhad znamená, že odhad $\hat{\alpha}$ je v průměru systematicky vyšší než skutečná hodnota parametru. Některé estimátory mohou být sice vychýlené, nicméně při dostatečně velké velikosti výběru konvergují ke skutečné hodnotě parametru. Takové estimátory nazýváme asymptoticky nestranné.

ÚKOL 7.3 Porovnejte vlastnosti estimátorů výběrového rozptylu s_x^2 a s'_x^2

$$s_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (7.4)$$

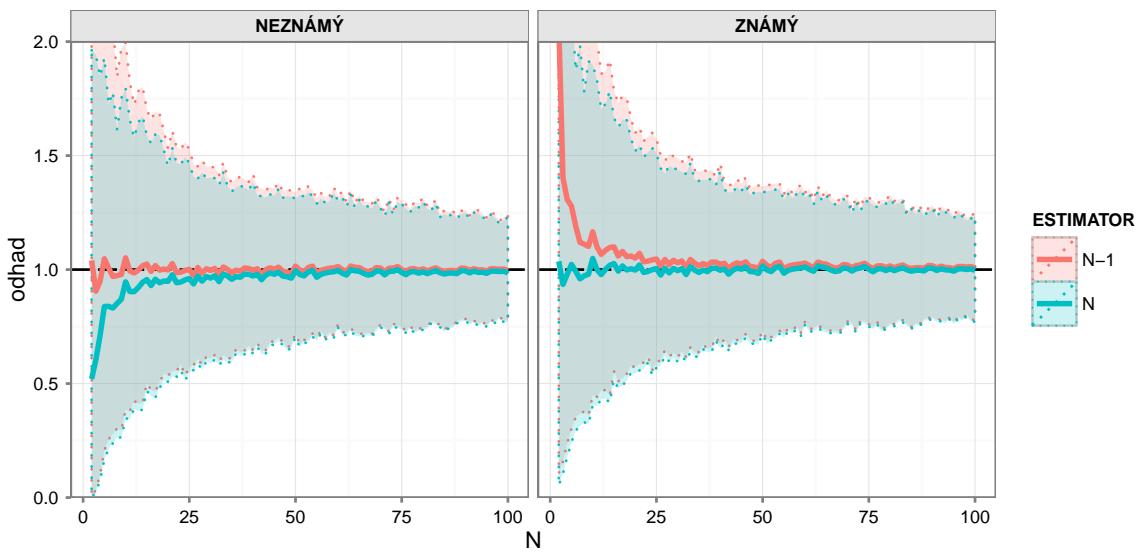
$$s'_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (7.5)$$

Použijte simulaci pro zvyšující se N . Jak se změní výsledky, budeme-li předpokládat, že střední hodnota veličiny X je známá?

Postup řešení:

- 1 Zvolte rozsah posuzovaných velikostí výběru $n_1:n_2$, např. $2:100$ a počet replikací $nsam$ pro každou hodnotu velikosti výběru, např. 100 . Interval $n_1:n_2$ není nutné samplovat celý, ale např. jen pro $N = 5, 10, 15 \dots$
- 2 Generujte $nsam$ -krát výběr o velikosti N z $n_1:n_2$ z vybraného rozdělení, např. $N(1, 1)$
- 3 Odhadněte rozptyl pro každou replikaci dle rovnic 7.4 a 7.5
- 4 Pokud ještě nebyla vybrána všechna N z $n_1:n_2$, běžte na bod 2.
- 5 Porovnejte odhadnuté rozptyly s očekávanou hodnotou - tj. s rozptylem rozdělení, ze kterého generujete výběr.

```
> require(data.table)
> require(ggplot2)
> require(reshape2)
>
> # 1)
> n1=2
> n2=100
> ns = seq(n1,n2, by = 1)
> nsam = 500
> rozptyl = 1
>
> # příprava proměnných pro ukládání výsledků
> res = array(NA, dim = c(length(ns), nsam, 2, 2),
+             dimnames = list(N = ns, SID = 1:nsam, ESTIMATOR=c('N-1','N'),
+               MEAN = c('NEZNÁMÝ', 'ZNÁMÝ')))
>
> # 2) - 4)
> for (N in ns){
+   for (i in 1:nsam){
+     x = rnorm(N, 0, sqrt(rozptyl))
+     seN = sum((x - mean(x))^2) # neznámý průměr
+     seZ = sum((x - 0)^2) # známý průměr
+     res[which(ns==N), i, ,1] = c(1/(N-1), 1/N) * seN
+     res[which(ns==N), i, ,2] = c(1/(N-1), 1/N) * seZ
+   }
+ }
> # 5)
> mres = data.table(melt(res))
> prum = mres[, list(q05 = quantile(value, .05),
+                      avg = mean(value), q95 = quantile(value, .95)), by = list(N, ESTIMATOR, MEAN)]
>
> ggplot(prum) +
+   theme_plot + # theme_plot je jen formátování pro tisk - není nutné
+   geom_ribbon(aes(x = N, ymin = q05, ymax = q95, fill = ESTIMATOR, col = ESTIMATOR),
+               alpha = 0.2, lty = 3) +
+   facet_grid(~MEAN) +
+   geom_hline(aes(yintercept = rozptyl), lty = 5) + xlab('N') + ylab('odhad') +
+   geom_line(aes(x = N, y = avg, col = ESTIMATOR), lwd = 1) +
+   coord_cartesian(ylim = c(0,2))
```



Obr. 7.3: Porovnání odhadů rozptylu dle rovnic 7.4 a 7.5 pro neznámou a známou střední hodnotu rozdělení.

Zatímco oba odhady jsou srovnatelně spolehlivé, je evidentní, že odhad dle rovnice 7.5 je vychýlený v případě neznámé střední hodnoty rozdělení a podobně odhad dle rovnice 7.4 je vychýlený v případě známé střední hodnoty rozdělení. Zároveň pro velká N jsou rozdíly minimální, oba odhady jsou asymptoticky nestranné.

□

Konzistence

Odhad je konzistentní, pokud platí, že střední kvadratická chyba odhadu se s rostoucí velikostí výběru blíží k nule, tj.

$$\lim_{N \rightarrow \infty} \text{MSE}(\alpha, \hat{\alpha}) = 0 \quad (7.6)$$

MSE lze přepsat i ve formě

$$\text{MSE}(\alpha, \hat{\alpha}) = [\text{B}(\hat{\alpha})]^2 + \text{V}(\hat{\alpha}) \quad (7.7)$$

aby byl estimátor konzistentní, musí být asymptoticky nestranný a mít konečný rozptyl.

V praxi hledáme takový estimátor, který je nestranný, co nejspolehlivější a konzistentní. Příkladem takovýchto estimátorů jsou odhady z kapitoly 4 - např. výběrové momenty a empirická distribuční funkce.

7.2 Maximálně věrohodný odhad

Někdy je obtížné nalézt nestranný odhad. V tom případě se často uvažuje tzv. maximálně věrohodný odhad. Maximálně věrohodný odhad je takový, který maximalizuje věrohodnostní funkci L pro výběr (x_1, \dots, x_N)

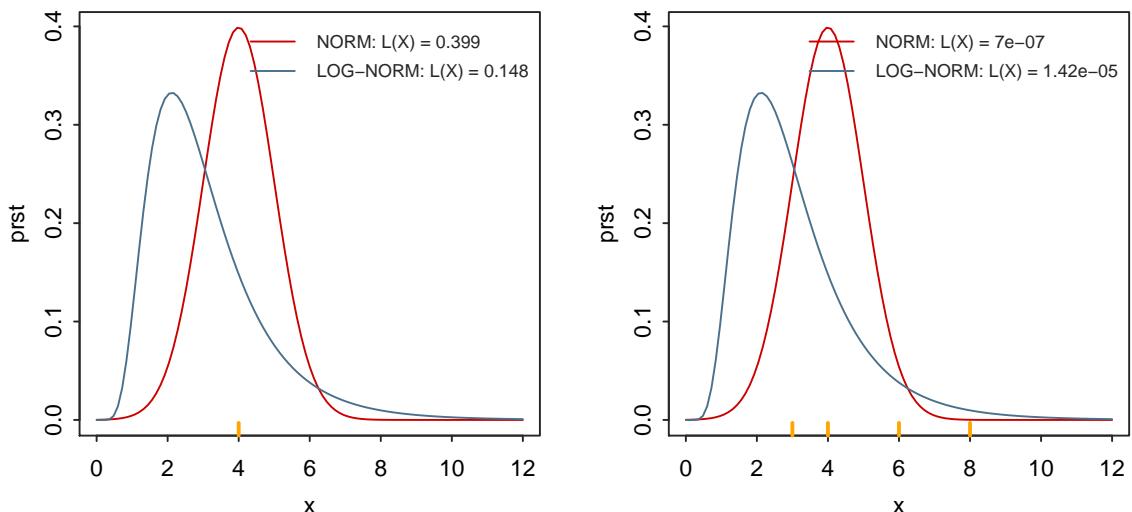
$$L(\theta, x_1, \dots, x_N) = p(x_1, \theta) \cdot p(x_2, \theta) \cdot \dots \cdot p(x_N, \theta) = \prod_i^N p(x_i, \theta) \quad (7.8)$$

Hodnota věrohodnostní funkce se vyčísluje pro nějaký pravděpodobnostní model - v nejjednoduší podobě nějaké rozdělení pravděpodobnosti a jeho parametry θ . Předpokládáme tedy například, že výběr pochází z normálního rozdělení a pomocí maximalizace věrohodnostní funkce hledáme jeho parametry, tedy střední hodnotu a rozptyl. Hodnota věrohodnostní funkce je maximální pro takový model, který vede s největší pravděpodobnosti k realizaci výběru X .

ÚKOL 7.4 Spočítejte hodnotu věrohodnostní funkce pro výběr $x = 4$, jako možné pravděpodobnostní modely použijte $N(4, 1)$ a $LN(1, 0.5)$. Vykreslete hustotu pravděpodobnosti těchto rozdělení a výběr ($x = 4$).

ÚKOL 7.5 Spočítejte hodnotu věrohodnostní funkce pro výběr $x = (3, 4, 6, 8)$ pro stejná rozdělení. Zkuste manuálně najít nejvhodnější vektor parametrů θ .

```
> par(c(list(mfrow = c(1,2)), par_plot))
> X = c(4)
> curve(dnorm(x, 4, 1), xlim = c(0, 12), xlab = 'x', ylab = 'prst', col='red3')
> curve(dlnorm(x, 1, .5), xlim = c(0, 12), add=TRUE, col='skyblue4')
> rug(X, lwd = 2, col='orange')
> lNORM = prod(dnorm(X, 4, 1))
> llogNORM = prod(dlnorm(X, 1, .5))
>
> legend('topright', paste(c('NORM: L(X) =', 'LOG-NORM: L(X) ='), round(c(lNORM, llogNORM), 3)), bty='n', col=
>
> X = c(3, 4, 6, 8)
> curve(dnorm(x, 4, 1), xlim = c(0, 12), xlab = 'x', ylab = 'prst', col='red3')
> curve(dlnorm(x, 1, .5), xlim = c(0, 12), add=TRUE, col='skyblue4')
> rug(X, lwd = 2, col='orange')
> lNORM = prod(dnorm(X, 4, 1))
> llogNORM = prod(dlnorm(X, 1, .5))
> legend('topright', paste(c('NORM: L(X) =', 'LOG-NORM: L(X) ='), round(c(lNORM, llogNORM), 7)), bty='n', col=
```



Obr. 7.4: Výsledek příkladu 7.4 (vlevo) a 7.5 (vpravo).

Maximálně věrohodný odhad je někdy zároveň nejlepším nestranným odhadem. Porovnání nejlepších nestranných a maximálně věrohodných odhadů vybraných rozdělení udává následující tabulka.

Tab. 7.1: Porovnání nejlepších nestranných a maximálně věrohodných odhadů vybraných rozdělení.

rozdělení	nejlepší nestranný odhad	maximálně věrohodný odhad
Normální rozdělení	$\hat{\mu} = \frac{1}{N} \sum_i x_i$ $\hat{\sigma}^2 = \frac{1}{N-1} \sum_i (x_i - \bar{x})^2$	$\hat{\mu} = \frac{1}{N} \sum_i x_i$ $\hat{\sigma}^2 = \frac{1}{N} \sum_i (x_i - \bar{x})^2$
Logaritmicko-normální rozdělení	$\hat{\mu} = \frac{1}{N} \sum_i \log x_i$ $\hat{\sigma}^2 = \frac{1}{N-1} \sum_i (\log x_i - \bar{x})^2$	$\hat{\mu} = \frac{1}{N} \sum_i \log x_i$ $\hat{\sigma}^2 = \frac{1}{N} \sum_i (\log x_i - \bar{x})^2$
Exponenciální rozdělení	$\hat{\lambda} = \frac{1}{N} \sum_i x_i$	$\hat{\lambda} = \frac{1}{N} \sum_i x_i$

7.3 Testování hypotéz

Statistický test lze definovat jako rozhodovací proces zjišťující, zda-li je daný výběr konzistentní s a priori formulovaným statistickým konceptem. Tento koncept se zpravidla nazývá nulová hypotéza a označuje se H_0 . V podstatě jsou možné pouze dva možné výsledky testování:

- H_0 se zamítá - tj. výběr obsahuje dostatek informací indikujících její neplatnost
- H_0 se nezamítá - tj. výběr neobsahuje dostatek informací indikujících její neplatnost

Statistickým testem tedy nemůžeme nikdy hypotézu potvrdit.

Samotný výsledek testování je náhodná veličina (protože je funkcí výběru), a proto při analýze jiné realizace náhodné veličiny X může být výsledek testování odlišný. V principu se tedy může stát, že nulovou hypotézu zamítneme, i pokud je správná. Chybu, kterou bychom se tímto dopustili (tj. zamítneme nulovou hypotézu, přestože je správná) označujeme jako chyba prvního druhu. Dopustit se můžeme i chyby, že nezamítneme nulovou hypotézu, ačkoliv správná není. Tuto chybu označujeme jako chyba druhého druhu. Je přirozené požadovat, aby pravděpodobnost obou těchto chyb byla co nejnižší. Nicméně zpravidla není možné zajistit, aby pravděpodobnost obou těchto chyb byla tak malá, jak bychom si přáli. Obvykle se trvá jen na požadavku, aby pravděpodobnost chyby prvního druhu byla rovna nějakému specifikovanému číslu α . Číslu α se říká hladina významnosti testu a volí se nejčastěji $\alpha = 0.1$, $\alpha = 0.05$ nebo $\alpha = 0.01$.

Mechanismus testu zpravidla pracuje s nějakou statistikou T a intervalem obsahujícím $(1 - \alpha) \cdot 100\%$ realizací veličiny T za předpokladu platnosti H_0 . Pak je hypotéza H_0 zamítnuta na hladině významnosti $(1 - \alpha) 100\%$, pokud pozorovaná hodnota statistiky $T = t$ se nalézá mimo tento interval. Hranice tohoto intervalu nazýváme kritické hodnoty. Pro jednoduché případy byla ve statistice zkonztruována pravidla, podle kterých nulovou hypotézu zamítáme. Použijeme-li těchto pravidel (a splníme-li předpoklady, pro které jsou zkonztruována), máme zajištěno, že se dopustíme chyby zamítnutí nulové hypotézy v případě její platnosti pouze s pravděpodobností nejvýše se rovnající hladině významnosti α . Zamítací pravidla jsou zkonztruována navíc tak, aby při zachování požadavku na pravděpodobnost chyby prvního druhu byla pravděpodobnost chyby druhého druhu co nejmenší (Jarušková, 2011).

Obecně požadujeme, aby statistický model, jenž je podkladem pro rozhodovací pravidlo, správně vystihoval charakter analyzované veličiny i způsob, jakým bylo pořízeno pozorování této veličiny. Pokud tomu tak není, je skutečná hladina významnosti testu odlišná od zvolené. Zároveň požadujeme,

aby rozhodovací pravidlo maximalizovalo pravděpodobnost, že hypotéza bude zamítnuta, pokud není pravdivá, tj. požadujeme, aby test měl co nejvyšší sílu (*power*).

Nulová hypotéza H_0 může být obecně zamítnuta z několika důvodů:

- Chybně zamítneme platnou nulovou hypotézu. Toto nelze nikdy vyloučit a souvisí to s definicí hladiny významnosti testu (tedy s chybou prvního druhu).
- Statistický model použitý pro konstrukci zamítacích pravidel není platný - tj. výběr nemusí odpovídat předpokladům (např. jednotlivá pozorování nejsou nezávislá) nebo nemusí mít předpokládané rozdělení (např. nemusí být symetrické kolem průměru). Test pak může zamítat H_0 podstatně častěji než dle zvolené hladiny významnosti.
- Správně jsme zamítlí neplatnou nulovou hypotézu.

Podobně rozhodnutí nezamítnout nulovou hypotézu H_0 může být způsobena několika důvody:

- H_0 může být neplatná, ale test nemá dostatek informací k indikaci její neplatnosti. Pravděpodobnost této chyby závisí na sile testu (pravděpodobnost chyby druhého druhu).
- Statistický model není konzistentní s analyzovanou náhodnou veličinou a zamítá H_0 jen zřídka. Důsledkem je slabý test.
- H_0 může být platná a test správně nemá dostatek informací k jejímu zamítnutí.

Nulová hypotéza má zpravidla podobu

$$H_0 : \theta = \theta_0 \quad (7.9)$$

kde θ je neznámý parametr, např. střední hodnota normálního rozdělení a testujeme, zda je hodnota tohoto parametru rovna nějakému zvolenému θ_0 . Alternativní hypotéza pak může mít v zásadě dvě podoby

1 oboustranná alternativa

$$H_1 : \theta \neq \theta_0 \quad (7.10)$$

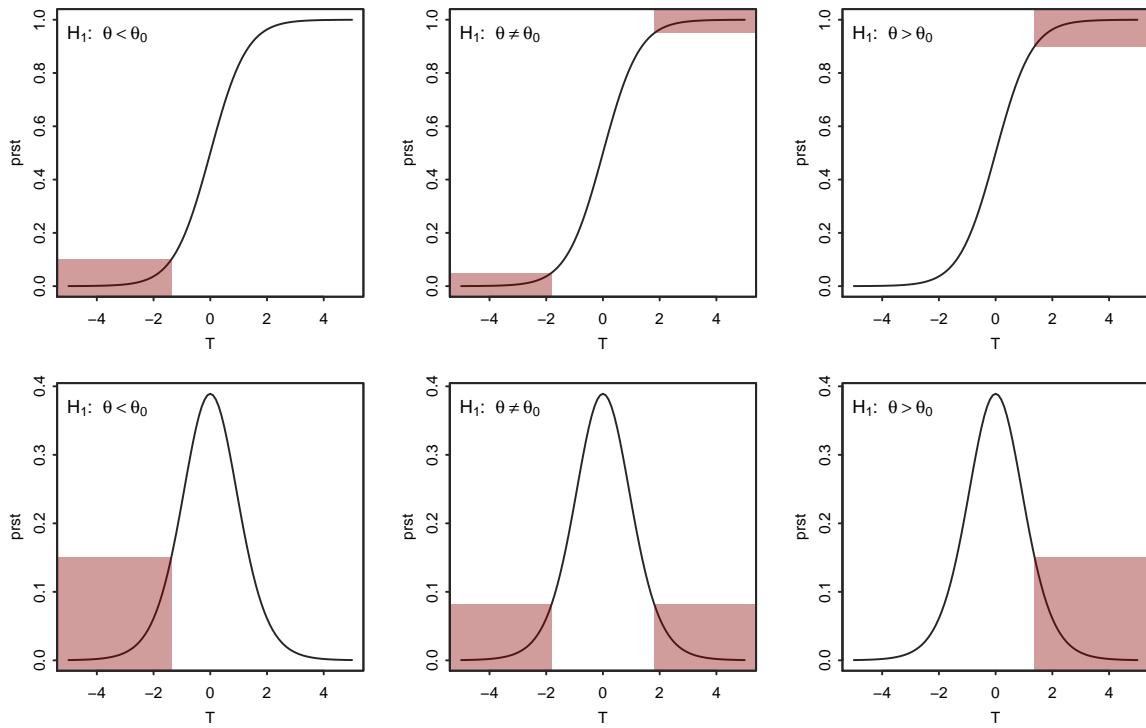
2 jednostranná alternativa

$$H_1 : \theta > \theta_0 \quad (7.11)$$

nebo

$$H_1 : \theta < \theta_0 \quad (7.12)$$

V prvním případě pak mluvíme o oboustranném testu, v druhém o jednostranném testu. V obou případech je nutné zajistit, aby pravděpodobnost, že testovací statistika nesplňuje nulovou hypotézu, byla vztažena k alternativní hypotéze.



Obr. 7.5: Ukázka distribuční funkce (horní řádek) a hustoty pravděpodobnosti (dolní řádek) testovací statistiky pro jednostranný a oboustranný test. Kritické oblasti jsou vyznačeny barevně.

Zamítací pravidla pro jednoduché nulové hypotézy lze nalézt v klasických učebnicích statistiky, např. Jarušková (2011); Puš (2011). Na tomto místě uvedeme jen několik příkladů. Asi nejpoužívanějším testem je tzv. t-test. Nejjednoduší aplikací t-testu je test hypotézy o střední hodnotě normálního rozdělení (μ), tedy např. testujeme hypotézu $H_0: \mu = \mu_0$ oproti alternativě $H_1: \mu \neq \mu_0$ nebo $H_1: \mu > \mu_0$. Zamítací pravidlo pro oboustrannou alternativu je definováno

$$\frac{|\bar{x} - \mu_0|}{s} \sqrt{N} > t_{\alpha/2}(N-1) \quad (7.13)$$

a pro jednostrannou alternativu ($\mu > \mu_0$)

$$\frac{\bar{x} - \mu_0}{s} \sqrt{N} > t_{\alpha}(N-1) \quad (7.14)$$

respektive

$$\frac{\bar{x} - \mu_0}{s} \sqrt{N} < -t_{\alpha}(N-1) \quad (7.15)$$

pro alternativu $\mu < \mu_0$, kde $t_{\alpha}(df)$ je 100α -procentní kvantil Studentova t-rozdělení s df stupni volnosti.

ÚKOL 7.6 Otestujte nulovou hypotézu, že střední hodnota rozdělení, ze kterého pochází níže uvedený výběr je rovna 10, větší než 10 a menší než 10. Hladinu významnosti testu uvažte 0.1.

```
> x = c(21, 12, 16, 4, 8, 21, -5, 2, 15, -5, 14, 29, 10, 7, 1, 8, -8, -18, 8, 13, 1, 13, 4, 3, -9, 23, 3, 22,
```

```

> N = length(x)
> alpha = 0.1
>
> # H0: mu = 10
>
> # H1a: mu != 10
> t = abs(mean(x)-10)/sd(x) * sqrt(N)
> t
[1] 2.932

> t > qt(alpha/2, N-1)
[1] TRUE

> # H1b: mu > 10
> t=(mean(x)-10)/sd(x) * sqrt(N)
> t
[1] -2.932

> t > qt(alpha, N-1)
[1] FALSE

> # H1c: mu < 10
> t=(mean(x)-10)/sd(x) * sqrt(N)
> t
[1] -2.932

> t < -qt(alpha, N-1)
[1] TRUE

```

V našem případě je možné H_0 zamítнуть na hladině významnosti 0.1 oproti alternativě H_{1a} (střední hodnota není rovna 10) a H_{1c} (střední hodnota je menší než 10). V případě testování oproti alternativě H_{1b} , že střední hodnota je vyšší než 10, nelze nulovou hypotézu zamítнуть.

□

Ve výše uvedeném případě jsme používali klasický přístup k testování, kdy hodnota testovací statistiky výběru je porovnávána s určitým kvantilem jejího teoretického rozdělení. Druhou možností je spočítat testovací statistiku a určit, jaká je pravděpodobnost, že testovací statistika nabývá této nebo extrémnější hodnoty vzhledem k alternativní hypotéze. Tato pravděpodobnost se nazývá p -hodnota. Nulovou hypotézu zamítáme, pokud je p -hodnota nižší, než námi zvolená hladina významnosti. p -hodnota nám zároveň říká na jaké hladině významnosti by bylo možné nulovou hypotézu zamítнуть, a nebo zjednoudušeně, jaká je pravděpodobnost, že nulová hypotéza platí „náhodou“.

Uvedená varianta t-testu se nazývá jednovýběrový t-test. Pomocí něj nicméně můžeme porovnávat i střední hodnoty dvou výběrů. Podoba testovací statistiky pak záleží na tom, zdali mají výběry stejný rozptyl a jestli jsou výběry párové, tj. například výběry vznikly měřením na stejně skupině subjektů před a po podání léku.

V Rku je k dispozici řada testů. K provedení t-testu slouží funkce `t.test`, která má následující argumenty:

Tab. 7.2: Argumenty funkce t.test.

argument	význam
x, y	vektory výběrů, v případě jednovýběrového testu se y nezadává
alternative = c("two.sided", "less", "greater")	typ alternativní hypotézy
mu	testovaná střední hodnota
paired = FALSE	je výběr párový, tzn. jsou dvojice hodnot ve výběru x a y závislé?
var.equal = FALSE	mají výběry stejný rozptyl?
conf.level	spolehlivost testu, tj. $1 - \alpha$

Testy z příkladu 7.6 by se provedly následujícím způsobem:

```
> # H0: mu = 10
>
> # H1a: mu != 10
> t.test(x, mu = 10, conf.level=0.9)
>
> # H1b: mu > 10
> t.test(x, mu = 10, conf.level=0.9, alternative='greater')
>
> # H1c: mu < 10
> t.test(x, mu = 10, conf.level=0.9, alternative='less')
```

Výstupem funkce t.test je výpis

```
> t.test(x, mu = 10, conf.level=0.9)
```

One Sample t-test

```
data: x
t = -2.932, df = 99, p-value = 0.004179
alternative hypothesis: true mean is not equal to 10
90 percent confidence interval:
 5.301 8.699
sample estimates:
mean of x
 7
```

kde t je výběrová testovací statistika, df jsou stupně volnosti a p-value je p -hodnota. V tomto případě můžeme nulovou hypotézu zamítнуть. Dále výpis udává alternativní hypotézu (alternative hypothesis) a požadovaný interval spolehlivosti (v příkladu 90 percent confidence interval) kolem výběrového průměru (mean of x).

ÚKOL 7.7 Dvouvýběrový t-test. Posuďte, zdali výběry x (z předchozího příkladu) a y mají stejnou střední hodnotu za předpokladu, že nejde o párové výběry a můžeme předpokladádat stejný rozptyl. Testujte oproti alternativě, že střední hodnoty jsou různé. Hladinu významnosti uvažujte 0.05.

```
> y = c(5, 15, 8, 1, 14, 21, 23, 15, 3, 16, -8, 0, 13, 27, -2, 2, 20, 20, -15, 2, 10, 11, 23, 15, 4, -2, 20, ...
>
> t.test(x, y, conf.level=0.95, var.equal=TRUE, paired=FALSE)
```

Two Sample t-test

```
data: x and y
t = -2.087, df = 198, p-value = 0.03821
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-5.6992 -0.1608
sample estimates:
mean of x mean of y
7.00      9.93
```

Hypotézu můžeme na zvolené hladině významnosti zamítnout. Výpis udává i 95% interval spolehlivosti okolo rozdílu průměru výběrů. Pokud by byla nulová hypotéza správná, měla by uvnitř ležet i 0.



Z definice hladiny významnosti očekáváme, že nulovou hypotézu zamítneme, přestože je správná s pravděpodobností α . To demonstруje následující příklad.

ÚKOL 7.8 Opakovaně generujte (např. 100krát) z normálního rozdělení dva výběry se stejnou střední hodnotou a stejným rozptylem a provedte t-test. Zjednočte, kolikrát test zamítl správnou hypotézu a porovnejte se zvolenou hladinou významnosti.

```
> alpha = 0.05
> pass = c()
> for (i in 1:100){
+     x = rnorm(50, 0, 5)
+     y = rnorm(50, 0, 5)
+     t = t.test(x, y, var.equal=TRUE, paired=FALSE)
+     pass[i] = t$p.value > alpha
+ }
> table(pass)/length(pass)

pass
FALSE  TRUE
0.05  0.95
```

Uvědomte si, že výsledek by měl odpovídat hladině významnosti pouze v průměru (jednotlivé realizace se mohou více nebo méně lišit). Měly bychom tedy simulaci opakovat a výsledky zprůměrovat.

ÚKOL 7.9 Zjistěte jaký vliv má předpoklad stejného rozptylu na výsledek?



Další důležitou vlastností je síla testu - tj. pravděpodobnost, že hypotéza bude zamítnuta pokud není pravdivá.

ÚKOL 7.10 Simulujte dva výběry podobně jako v příkladu 7.8 s tím, že výběry budou mít různou střední hodnotu a stejný rozptyl. Vyhodnoťte pravděpodobnost, že nulová hypotéza je zamítnutá.

```
> alpha = 0.05
> pass = c()
> for (i in 1:100){
+     x = rnorm(50, 2, 5)
+     y = rnorm(50, 0, 5)
+     t = t.test(x, y, var.equal=TRUE, paired=FALSE)
```

```

+         pass[i] = t$p.value>alpha
+ }
> table(pass)/length(pass)

pass
FALSE  TRUE
0.49  0.51

```

Je evidentní, že náš výsledek závisí na velikosti rozdílu mezi středními hodnotami a jeho poměru k rozptylu výběrů. To můžeme znázornit simulací z rozdělení s různými rozdíly středních hodnot:

```

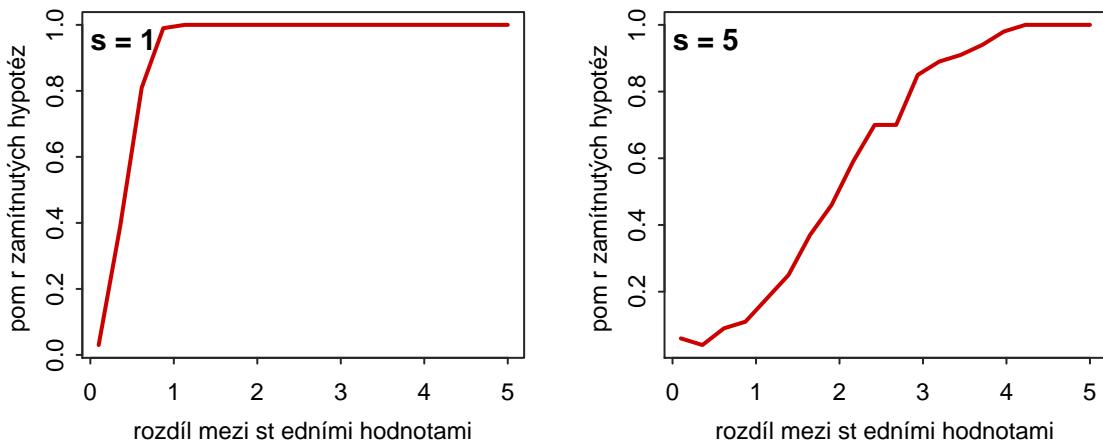
> alpha = 0.05
> dif = seq(0.1, 5, length=20) # samplovaný rozdíl mezi středními hodnotami
> for (v in c(1, 5)){ # směrodatná odchylka je 1 a 5
+ npass = c()
+ pass = c()

```

```

+ for (d in dif){
+   for (i in 1:100){
+     x = rnorm(50, d, v)
+     y = rnorm(50, 0, v)
+     t = t.test(x, y, var.equal=TRUE, paired=FALSE)
+     pass[i] = t$p.value>alpha
+   }
+ npass= c(npass, length(which(pass==FALSE))/length(pass))
+ }
+ plot(dif, npass, type='l', xlab='rozdíl mezi středními hodnotami', ylab='poměr zamítnutých hypotéz', col='red')
+ title(main = paste(' s = ', v), line=-1, adj = 0)
+ }

```



Obr. 7.6: Závislost poměru zamítnutých testů na rozdílu mezi středními hodnotami.

□

8 Volba modelu a vyhodnocení shody

Verifikace numerické předpovědi je rozšířena jako vhodný způsob získávání zpětné vazby na výsledky vyhodnocení úspěšnosti. Výsledky verifikace mohou být použity pro hodnocení a posouzení přesnosti předpovědi nebo porovnání kvality modelů. Lze tímto způsobem také identifikovat určité nedostatky modelu za účelem jejich nápravy, případně jakým směrem by se měl zaměřit budoucí vývoj. Na druhou stranu je možné určit výhody a silné stránky modelu, tzn. například situace, ve kterých je předpověď velice přesná (Jolliffe a Stephenson, 2012; Vokoun, 2014).

8.1 Data pro vyhodnocení

Data pro vyhodnocení událostí lze rozdělit do dvou kategorií, a to na: *binární události a kontinuální veličiny*.

8.1.1 Binární události

Mezi binární, neboli dichotomní jevy, lze zařadit mnoho z pozorovaných meteorologických událostí, jako například dešť, povodně, mlha a jiné. Tyto jevy buďto nastaly, nebo ne. Tento druh předpovědi je také někdy nazýván jako „*ano/ne*“ předpověď. Existují tedy čtyři různé kombinace výsledných výstupů, které mohou nastat. Výsledky jsou nejčastěji prezentovány pomocí kontingenční tabulky. Způsob zápisu úspěšných prognóz je *hit* (událost byla předpovězena a nastala) nebo *correct rejection* (událost nebyla předpovědena a nenastala) a neúspěšných prognóz jako *false alarm* (událost byla předpovězena a nenastala) nebo *miss* (událost nebyla předpovězena a nastala).

Frequency Bias (BIAS): poměr mezi počtem předpovědí výskytu události a skutečným počtem výskytu události. Vzhledem k tomu, že pracuje pouze četnostmi, z výsledku nezjistíme přesnost předpovědi. Smyslem je určit, zda model podhodnocuje nebo nadhodnocuje předpověď. Nejlepší skóre je 1.

$$BIAS = \frac{hits + false alarms}{hits + misses} \quad (8.1)$$

Probability of Detection (POD): bývá také často označován jako *hit rate* (*H*). Vyjadřuje podíl správně předpokládaných událostí, na které se zaměřuje. Naopak úplně opomíjí false alarms. Nejlepší skóre je 1.

$$POD = \frac{hits}{hits + misses} \quad (8.2)$$

False Alarm Ratio (FAR): zjišťuje k jakému podílu předpovídaných „yes“ ve skutečnosti nedošlo. Měl by být využit v souvislosti s POD. FAR je vzorek odhadu podmíněný pravděpodobnosti falešného

poplachu tím, že předpovídaná událost nenastala. Nejlepší skóre je 0.

$$FAR = \frac{\text{false alarms}}{\text{hits} + \text{false alarms}} \quad (8.3)$$

False alarm rate (F): popisuje podíl nenaplněných předpovědí, které byly předpovídány. Jindy se také nazývá Probability Of False Detection (POFD). Ideální výsledek je 0 a lze jej vylepšit snížením počtu vydaných „yes“ předpovědí.

$$F = \frac{\text{false alarms}}{\text{correct rejections} + \text{false alarms}} \quad (8.4)$$

Threat score (TS): snaží se postihnout, do jaké míry korespondují „yes“ předpověď s „yes“ měřeními. Často je také uváděn jako Critical Success Index (CSI). Nerozlišuje zdroj chyby předpovědi a postihuje stejnou mírou *misses* a *false alarms*. TS statistika je široce používána pro hodnocení odhadu vzácných událostí, protože k výpočtu není třeba znát frekvenci správných zamítnutí (*correct rejections*).

$$TS = \frac{\text{hits}}{\text{hits} + \text{misses} + \text{false alarms}} \quad (8.5)$$

Equitable Threat Score (ETS): stejně jako TS, upraveno pro počet shod, které byly náhodně vybrány, kde:

$$\text{hits}_{\text{random}} = \frac{1}{N} (\text{pozorovaných } "yes" \times \text{předpovídaných } "yes") \quad (8.6)$$

ETS je často používána při verifikaci srážek v modelech. Jako přednost je uváděna rovnovážnost hodnocení, která umožňuje provádět spravedlivá porovnání mezi různými režimy. Ideální skóre je 1.

$$ETS = \frac{\text{hits} - \text{hits}_{\text{random}}}{\text{hits} + \text{misses} + \text{false alarms} - \text{hits}_{\text{random}}} \quad (8.7)$$

Hanse and Kuipers discriminant (HK): vypočítává se rozdílem POD hodnocení a F hodnocení a měří schopnost předpovědi odlišit pozorované „yes“ případy od „no“ případů. K výpočtu jsou využity všechny čtyři možné výsledky z kontingenční tabulky. Ideální skóre je 1.

$$HK = POD - F \quad (8.8)$$

8.1.2 Kontinuální veličiny

Mezi *kontinuální veličiny* se řadí reálné spojité (kontinuální) skalární veličiny, jako například teplota, tlak nebo srážky. Kontinuální reálné veličiny jsou běžně výsledkem numerické předpovědi a kategorické předpovědi (předchozí kapitola) jsou často získávány pomocí prahových hodnot pro spojité veličiny. Příkladem může být teplotní řada měřených a předpovídaných hodnot. Mimo souhrnných skóre lze použít pro verifikaci také řadu grafických posouzení (bodové a krabicové grafy).

Mean Absolute Error (MAE): nevýhodou střední (systematické) chyby je ta, že záporné chyby (odchyly) se kompenzují těmi kladnými. To znamená, že i dobře hodnocená předpověď nemusí být tak

přesná, jak by se zdálo. Tomuto jevu se snaží předejít právě MAE, která je definována jako průměr absolutních hodnot. Z tohoto důvodu však nelze určit směr odchylky. Rm_i představuje předpovídánou hodnotu a R_i naměřenou hodnotu. Ideální skóre je 0.

$$MAE = \frac{1}{n} \sum_{i=1}^n |RM(i) - R(i)|^2, \quad (8.9)$$

Mean Squared Error (MSE): patří mezi jedno z nejpoužívanějších hodnocení. Díky druhé mocnině je MSE daleko více citlivé na větší odchylky předpovědi, než MAE. Výsledkem je suma druhých mocnin rozdílu mezi předpověď a měřením.

$$MSE = \frac{1}{n} \sum_{i=1}^n (RM(i) - R(i))^2, \quad (8.10)$$

Root Mean Square Error (RMSE): stejně jako MSE klade důraz na větší chyby. RMSE měří průměrný rozsah chyby (výsledná odchylka je na rozdíl od MSE ve stejně jednotce jako vstupní hodnoty), z čehož plyne, že stejně jako v případě MSE je ideální skóre 0.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (RM(i) - R(i))^2}, \quad (8.11)$$

V prostředí R je možné využít pro vyhodnocení balík hydroGOF.

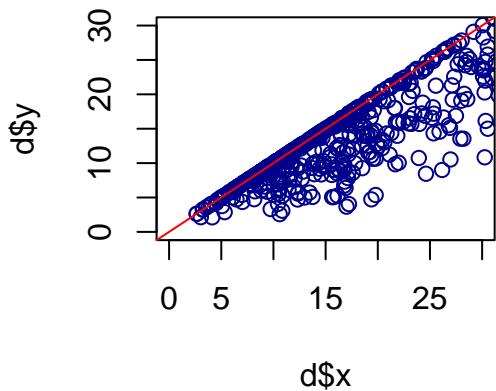
ÚKOL 8.1 Vypočítejte kritéria MAE, MSE a RMSE pro pozorované a modelované průtoky

Scatter plot: jednoduchý graf stavějící proti sobě předpovídáné a měřené hodnoty. V případě ideální shody jsou všechny body seřazeny na 45° diagonále s počátkem v bodu 0. Z grafu lze například vyčíst, zda předpověď nadhodnocuje nebo podhodnocuje, odlehle hodnoty nebo tendenci.

Quantile plot: v grafu jsou vyneseny proti sobě vybrané kvantily měřených a predikovaných hodnot. Matematicky se jedná o bodový graf podmíněný kvantily měřených a predikovaných hodnot pro vybranou pravděpodobnost. Nejlepšího výsledku je opět dosaženo, pokud body leží na nebo v těsné blízkosti 45° diagonály. Pokud se nacházejí dále od diagonály, ale stále pohromadě, je potřeba nakalibrovat model. Závažnější problémem je větší vzdálenost v grafu například mezi body 0,25 kvantilu a 0,75 kvantilu.

ÚKOL 8.2 Nakreslete scatter a quantile plot pro pozorované a modelované průtoky

```
> require(data.table)
>
> # nahrání dat
> dta = fread("../0230_bilance.dat")
>
> d <- data.table(x = dta$V5, y = dta$V6)
> plot(d$x, d$y, xlim = c(0, 30), ylim = c(0, 30), type = "p", col = "dark blue")
> abline(0, 1, col = "red")
```



Obr. 8.1: Scatter plot pro modelované průtoky.

Mean Absolute Percentage Error (MAPE): kritérium absolutních hodnot relativních odchylek.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|RM(i) - R(i)|}{R(i)} \quad (8.12)$$

Pro vyhodnocení lze také použít kritérium **Nash-Sutcliffe efficiency** (NS) nebo **logarithmic Nash-Sutcliffe** (LNNS).

$$NS = 1 - \frac{\sum_{i=1}^n (RM(i) - R(i))^2}{\sum_{i=1}^n (R(i) - \bar{R})^2} \quad (8.13)$$

$$LNNS = 1 - \frac{\sum_{i=1}^n (\ln RM(i) - \ln R(i))^2}{\sum_{i=1}^n (\ln R(i) - \bar{\ln R})^2}, \quad (8.14)$$

$$\bar{\ln R} = \frac{1}{n} \sum_{i=1}^n \ln R(i). \quad (8.15)$$

ÚKOL 8.3 Vypočítejte kritérium Nash-Sutcliffe efficiency.

ÚKOL 8.4 Jaký je rozdíl mezi koeficientem determinace a Nash-Sutcliffe efficiency? K čemu se tato kritéria využívají?

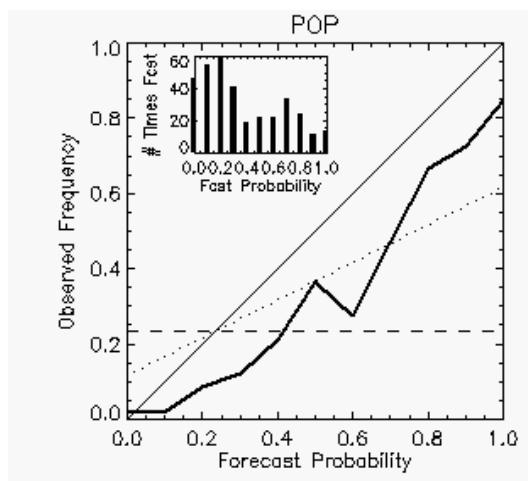
8.2 Vyhodnocení pravděpodobnostních modelů

Předchozí metody byly zaměřeny na hodnocení dat ve formě jedné hodnoty (z kontinua) nebo diskrétní kategorie. Pro pravděpodobnostní předpověď je typická prezentace dat v určitých intervalech nebo třídách (kategoriích). To znamená, že na základě očekávané pravděpodobnosti výskytu předpovídané

události je určena pravděpodobnostní hodnota jevu P v rozmezí 0 až 1. Pravděpodobnostní předpověď může být také založena na souboru několika deterministických předpovědí platných pro stejný čas, které jsou nezávislé a podléhají náhodnému procesu. Odhad pravděpodobnosti prognózované události poskytuje podíl předpovědí predikujících určitou událost ze všech uvažovaných předpovědí. Tato metoda je známá jako ansámblová předpověď a je stále rozšířenější (Jolliffe a Stephenson, 2012; Vokoun, 2014). Na rozdíl od kategorické předpovědi není výstupem předpovědi například „výskyt deště v určitý časový interval: „yes“, nýbrž kupříkladu tvrzení, že dešť se vyskytne s 40% pravděpodobností. V tomto případě je třeba provádět verifikaci pro jednotlivé míry pravděpodobnosti (40%). Stejně tak lze v pravděpodobnostní předpovědi hodnotit také binární události, a to opět principem přiřazení hodnot, zda k události došlo (1), nebo se nevyskytla (0).

K hodnocení pravděpodobnostní předpovědi lze kromě empirických metrik využít také grafické zpracování výsledků. Jednou z možností je *spolehlivostní křivka* (Reliability Curve-RC) nebo *křivka relativní operační charakteristiky* (Relative Operating Characteristic – ROC).

Spolehlivostní křivka: znázorňuje přesnost předpovědí v přímé závislosti k naměřeným hodnotám (obr. 8.2). Spolehlivostní křivku představuje plná čára, získaná vnesením hodnot $f(q)$ frekvence události proti q pravděpodobnosti předpovědi. V ideálním případě se spolehlivostní křivka překrývá s diagonální linií. Horizontální přerušovaná linie představuje klimatologickou frekvenci popisované události a tečkovaná linie vedoucí středem mezi diagonální a horizontální linií značí tzv. „no skill“ ve vztahu ke klimatologii. To znamená, že pouze hodnoty nad touto tečkovanou čarou přikládají vyšší schopnost předpovědi, než pokud bychom vycházeli z klimatologických podkladů. V zobrazeném příkladě se nachází spolehlivostní křivka napravo od diagonály. To znamená, že předpověď byla nadhodnocena. Histogram v levém horním rohu ukazuje četnost jednotlivých hodnot pravděpodobností, které byly predikovány.



Obr. 8.2: Spolehlivostní křivka

Relativní Operační Charakteristika (ROC): křivku dostaneme vykreslením výsledků charakteristik POD proti F pro události definované pomocí řady rozhodovacích prahových hodnot. Prahovou hodnotou bývá pravděpodobnost a „yes“ událost nastane, pokud pravděpodobnost předpovědi přesáhne prahovou hranici (Ebert, 2008). Křivka v případě přesné předpovědi začíná v levé spodní části a následuje osu Y směrem do levého horního rohu ($F=0$). Dále následuje horní osu směrem k pravému hornímu rohu ($POD=1$). Výsledkem je plocha vzniklá pod křivkou, kdy ideální hodnota je 1, „no

skill“ hodnota je 0,5.

ÚKOL 8.5 Vykreslete ROC, pro vykreslení použijte data z balíku pROC

```
> require(pROC)
> data(aSAH)
> # Syntaxe
> plot.roc(aSAH$outcome, aSAH$s100b)
```

Call:

```
plot.roc.default(x = aSAH$outcome, predictor = aSAH$s100b)
```

Data: aSAH\$s100b in 72 controls (aSAH\$outcome Good) < 41 cases (aSAH\$outcome Poor).
Area under the curve: 0.731

```
> head(aSAH$outcome)
```

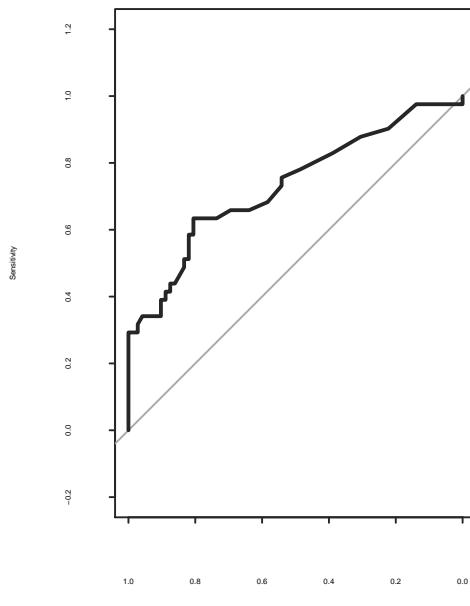
```
[1] Good Good Good Good Poor Poor
```

Levels: Good Poor

```
> head(aSAH$s100b)
```

```
[1] 0.13 0.14 0.10 0.04 0.13 0.10
```

```
> plot(rocobj, print.auc = TRUE, auc.polygon = TRUE, grid = c(0.1, 0.2), grid.col =
  c("green",
+   "red"), max.auc.polygon = TRUE, auc.polygon.col = "blue", print.thres = TRUE)
```



Obr. 8.3: Křivka ROC

Brier Score (BS): Brier navrhl několik metod hodnocení kvantitativních pravděpodobnostních binárních prognóz, založených na kvadratickém bodování. Jednou z těchto metod je právě BS, kde N je počet realizací předpovídáního procesu, přes které je validace prováděna (Jolliffe a Stephenson, 2012; Vo-

koun, 2014). Pro každou realizaci i , existuje pravděpodobnost výskytu predikované události p_i , a stejně tak hodnota o_i nabývající hodnoty 1 nebo 0, podle toho, zda se událost vyskytla nebo ne. BS poměřuje střední kvadratickou pravděpodobnostní chybu při N událostech. Ideální skóre je 0 (Ebert, 2008).

$$BS = \frac{1}{N} \sum_{i=1}^N (p_i - o_i)^2 \quad (8.16)$$

Brier Skill Score (BSS): na rozdíl od BS je BSS kladně orientováno, tedy čím vyšší hodnota, tím lepší přesnost předpovědi a navíc bere v úvahu klimatologickou frekvenci (Bref). Stejně jako BS je vhodnější pro delší datové sady a především v případě výskytu extrémní události, je potřeba více dat pro dosažení relevantních výsledků.

$$BSS = 1 - \frac{BS}{BS_{ref}} \quad (8.17)$$

Ranked Probability Score (RPS): měří sumu kvadratický rozdílů v kumulativním kvadratickém prostoru pro multi-kategoriální pravděpodobnostní předpověď. $CDF_{y,m}$ je kumulativní pravděpodobnost předpovědi, že bude překročena definovaná hranice pro m . Pokud je překročena, o_m nabývá hodnoty 1, pokud ne, 0. M představuje počet kategorií předpovědi.

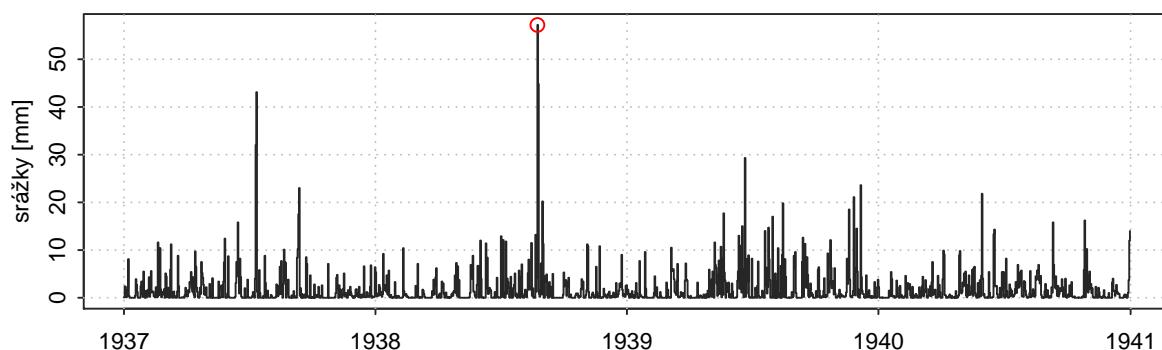
9 Analýza extrémů

9.1 Definice extrémů

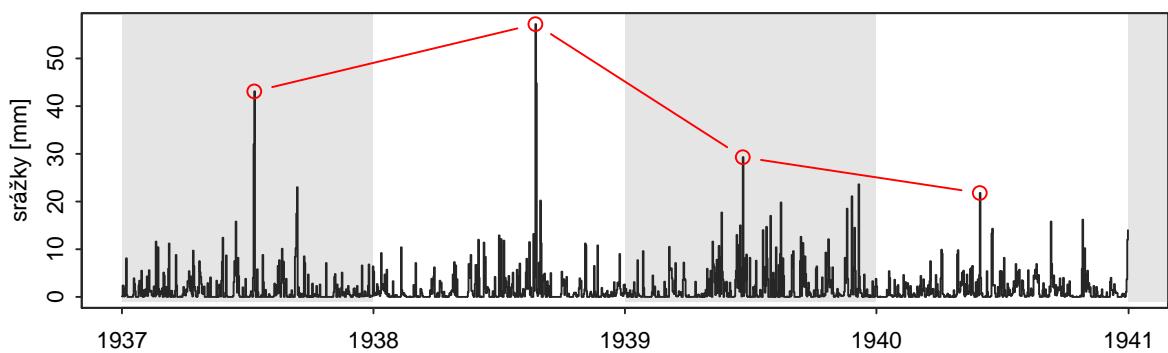
Extrémy v časové řadě mohou být definovány několika způsoby:

- 1 Absolutní maximum - nejvyšší hodnota v dané časové řadě
- 2 Bloková maxima (*Block maxima*)
Časová řada dané veličiny je rozdělena do pravidelných bloků (např. let, měsíců atp.). Uvažována jsou potom maxima jednotlivých bloků (tedy roční, měsíční maxima atp.).
- 3 Nadprahové hodnoty (*Peaks over threshold*)
Pro danou veličinu je zvolena prahová hodnota - zpravidla nějaký vysoký kvantil jejího rozdělení. Uvažovány jsou potom hodnoty nad tímto prahem.

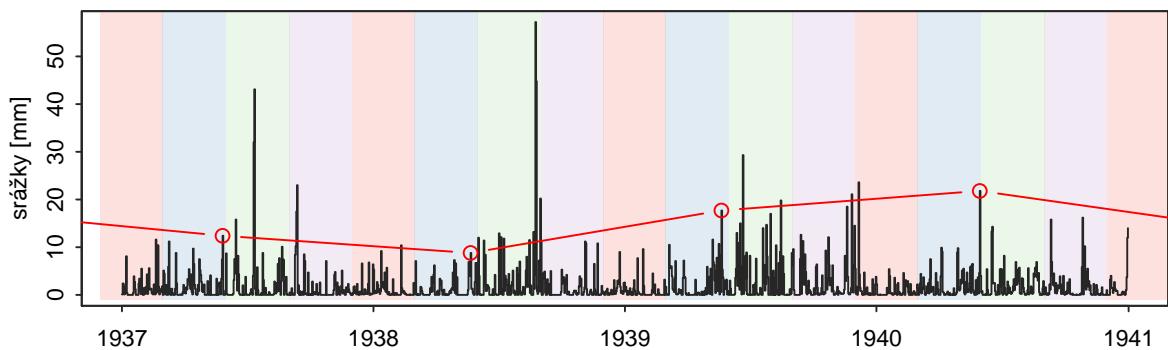
Ilustrace těchto přístupů ukazují obrázky 9.1 – 9.5. Každá z uvedených metod má své výhody a nevýhody. Rozdělení blokových maxim konverguje za relativně slabých předpokladů k zobecněnému rozdělení extrémních hodnot (Generalized extreme value distribution, GEV). Nevýhodou je, že analýza je založena jen na zlomku dat, což vede k značné nejistotě odhadů jeho parametrů. Na druhé straně modelování extrémů jako nadprahových hodnot je spojeno s problematikou volby prahu. Je-li práh příliš vysoký, extrémů je málo, což vede k zvyšování chyby odhadu, je-li práh příliš nízký, extrémů je větší počet, nicméně hrozí nebezpečí, že rozdělení těchto hodnot nebude mít požadované asymptotické vlastnosti. Pokud modelovaná veličina vykazuje sezónní cyklus, měl by se práh během roku měnit, v případě nestacionární řady by navíc měl být funkcí času či jiného indikátoru nestacionarity.



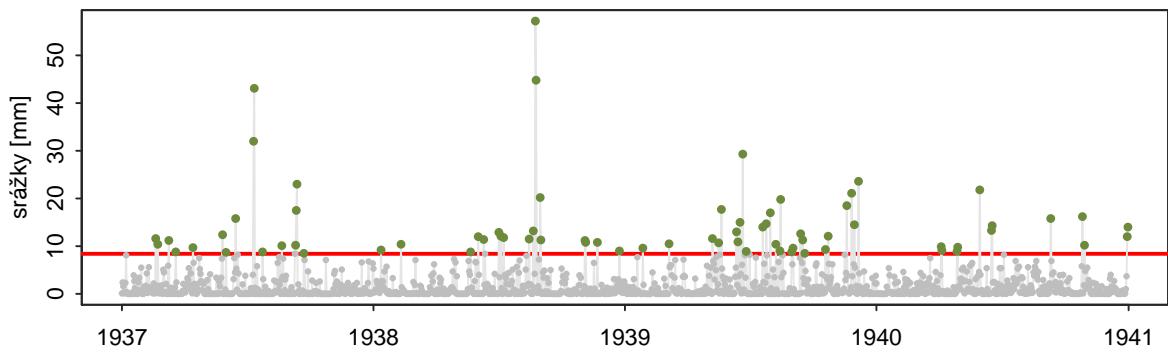
Obr. 9.1: Absolutní maximum.



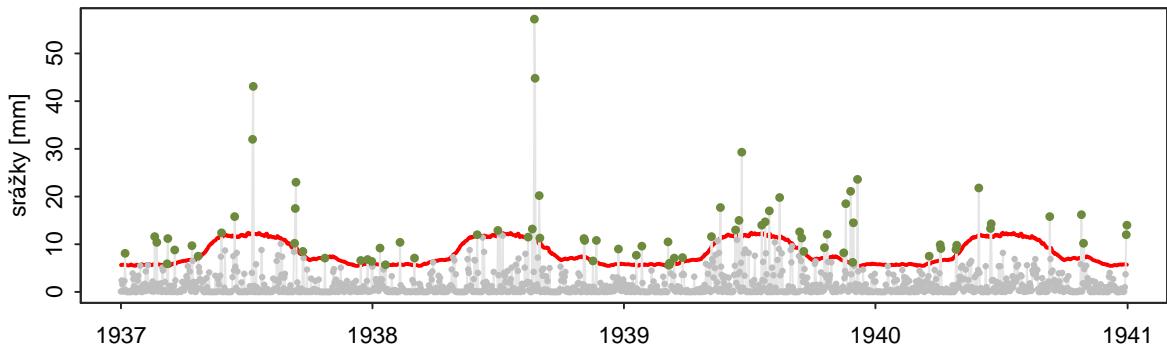
Obr. 9.2: Bloková maxima - roční.



Obr. 9.3: Bloková maxima - sezónní.



Obr. 9.4: Nadprahové hodnoty - stacionární práh.



Obr. 9.5: Nadprahové hodnoty - proměnný práh.

S analýzou extrémů je úzce spjat koncept doby opakování T . Doba opakování T je průměrná doba, za jakou je hodnota určité veličiny dosažena nebo překročena. Zpravidla nás zajímá hodnota (např. odtok) pro zvolenou dobu opakování (např. 10 let, 100 let) - zejména pro návrhové účely. Pravděpodobnost pro dobu opakování T je možno vypočítat jako

$$p = 1 - \frac{1}{T} \quad (9.1)$$

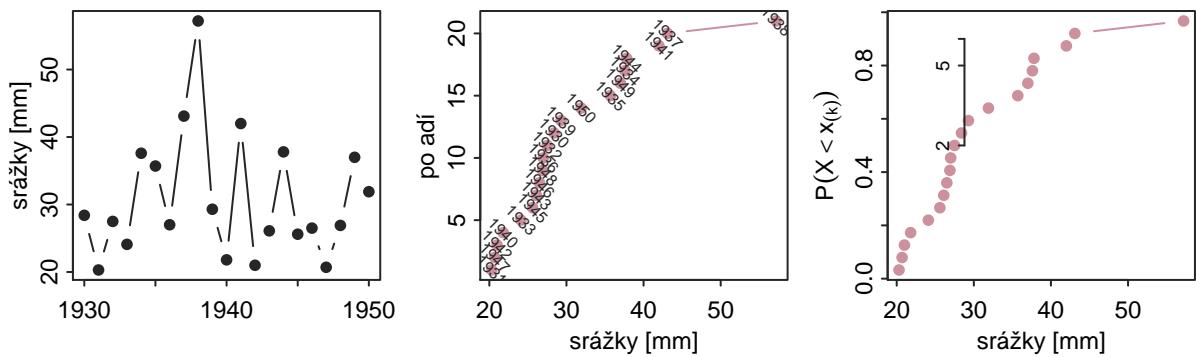
neboli k pravděpodobnosti p náleží doba opakování

$$T = \frac{1}{1 - p}, \quad (9.2)$$

kde $p = P(X \leq x)$ je hodnota distribuční funkce.

ÚKOL 9.1 Pro data definovaná v následujícím kódu určete na základě empirické distribuční funkce 10letou srážku.

```
> par(mar=c(2.5, 2, .5, .5), mgp = c(1, .2, 0))
> # vektor maxim
> MX
[1] 28.4 20.3 27.5 24.1 37.6 35.7 27.0 43.1 57.2 29.3 21.8 42.0 21.0 26.1
[15] 37.8 25.6 26.5 20.7 26.9 37.0 31.9
> # a roky, kdy byla maxima pozorována
> rok
[1] 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943
[15] 1944 1945 1946 1947 1948 1949 1950
> # zdrojová data
> plot(rok, MX, type='b', xlab='', ylab='srážky [mm]', pch=20)
>
> # pořádková statistika
> plot(sort(MX), 1:length(MX), ylab='pořadí', xlab='srážky [mm]', col='pink3', pch=20, type='b')
> text(MX, rank(MX), rok, cex=.7, srt=-45)
>
> # empirická distribuční funkce
> P = 1 - 1/c(2,5,10)
> p = (rank(MX)-0.3)/(length(MX)+0.4)
> plot(sort(p), sort(MX), ylab=expression(P(X<x[(k])), xlab='srážky [mm]', col='pink3', pch=20, type='b')
> axis(2, at=P, label=c(2,5,10), line=-2, cex.axis=0.8)
```



Obr. 9.6: Časová řada maxim (vlevo). Pořádková statistika (uprostřed) a empirická distribuční funkce (vpravo). Druhá osa y ukazuje 2, 5 a 10letou srážku.

□

9.2 Teoretické modely extrémů

GEV rozdělení

Za určitých (poměrně obecných) předpokladů mají bloková maxima jedno z následujících rozdělení:

- Gumbelovo

$$F(x) = \exp \left\{ -\exp \left[-\frac{x-\xi}{\alpha} \right] \right\} \quad (9.3)$$

- Fréchetovo

$$F(x) = \exp \left[- \left(\frac{x-\xi}{\alpha} \right)^{-\kappa} \right] \quad (9.4)$$

- obrácené Weibullovo

$$F(x) = \exp \left\{ - \left[-\frac{x-\xi}{\alpha} \right]^\kappa \right\} \quad (9.5)$$

tato tři rozdělení je možno sumarizovat pomocí tzv. zobecněného rozdělení extrémních hodnot (Generalized Extreme Value distribution - GEV):

$$F(x) = \exp \left\{ - \left[1 + \kappa \left(\frac{x-\xi}{\alpha} \right) \right]^{-\frac{1}{\kappa}} \right\}, \quad \kappa \neq 0 \quad (9.6)$$

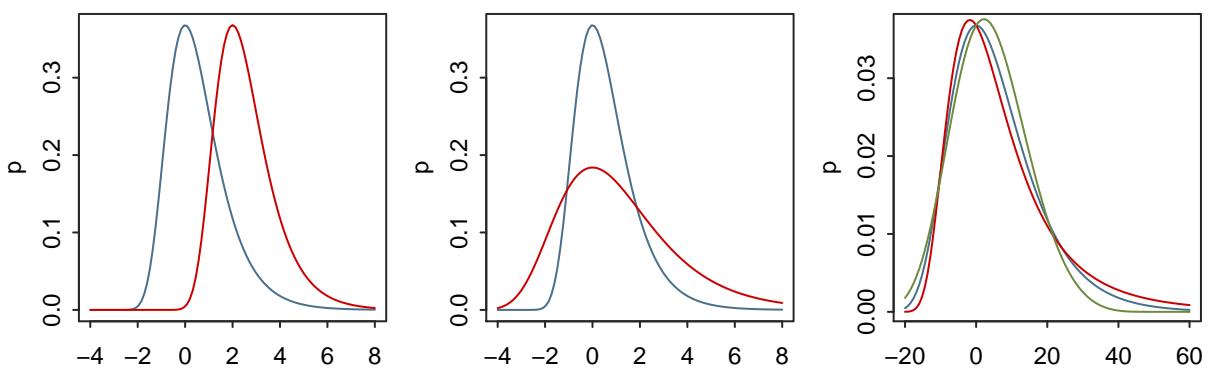
$$F(x) = \exp \left\{ - \exp \left[- \left(\frac{x-\xi}{\alpha} \right) \right] \right\}, \quad \kappa = 0 \quad (9.7)$$

kde ξ – location parametr – určuje polohu rozdělení, α – scale parametr – určuje variabilitu rozdělení a κ – shape parametr – určuje chování chvostu rozdělení:

- $\kappa = 0$ - Gumbelovo rozdělení
- $\kappa > 0$ - Fréchetovo rozdělení

- $\kappa < 0$ - obrácené Weibullovo rozdělení

```
> par(mar = c(1.5, 2, .5, .5))
> require(evd)
>
> curve(dgev(x, loc=0, sca=1, sha=0), xlim = c(-4, 8), col = 'skyblue4', ylab='p', xlab='MX')
> curve(dgev(x, loc=2, sca=1, sha=0), xlim = c(-4, 8), col='red3', add=TRUE)
>
> curve(dgev(x, loc=0, sca=1, sha=0), xlim = c(-4, 8), col = 'skyblue4', ylab='p', xlab='MX')
> curve(dgev(x, loc=0, sca=2, sha=0), xlim = c(-4, 8), col='red3', add=TRUE)
>
> curve(dgev(x, loc=0, sca=10, sha=0), xlim = c(-20, 60), col = 'skyblue4', ylab='p', xlab='MX')
> curve(dgev(x, loc=0, sca=10, sha=0.2), xlim = c(-20, 60), col='red3', add=TRUE)
> curve(dgev(x, loc=0, sca=10, sha=-0.2), xlim = c(-20, 60), col='darkolivegreen4', add=TRUE)
```



Obr. 9.7: GEV rozdělení lišící se (vlevo) polohou, (uprostřed) variabilitou, (vpravo) tvarem – zelená ukazuje rozdělení s $\kappa = -0.1$, červená s $\kappa = 0.1$.

GPA rozdělení

Nevýhodou výše uvedeného modelu je, že uvažuje pouze maximální událost a další, které mohou být téměř stejně veliké, ignoruje. To je možno obejít alternativní definicí extrému, a to jako hodnotou nad určitým prahem u . Pro veličinu X mají za určitých předpokladů nadprahové hodnoty $Y = X - u$ GPA rozdělení - tj. dvouparametrické rozdělení s parametrem tvar ξ a škála σ .

$$P(Y \leq y | Y \geq 0) = G(y) = 1 - \left(1 + \frac{\xi y}{\sigma}\right)^{-1/\xi}, \quad \xi \neq 0 \quad (9.8)$$

$$P(Y \leq y | Y \geq 0) = G(y) = 1 - \exp\left(\frac{y}{\sigma}\right), \quad \xi = 0 \quad (9.9)$$

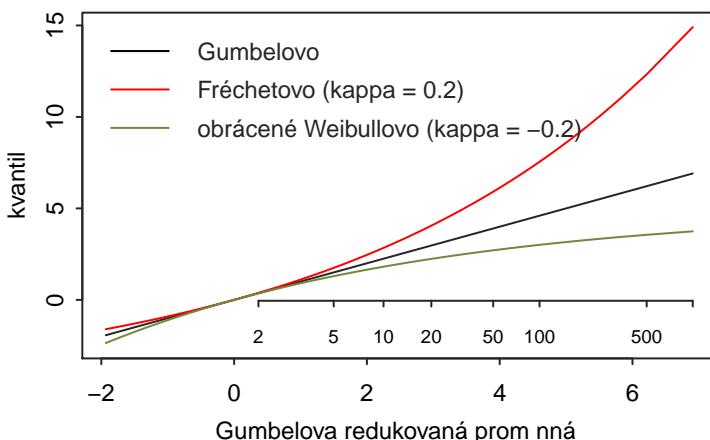
Problémem je zvolit správný prah (tj. dostatečně vysoký, aby platily předpoklady modelu a zároveň co nejnižší, aby bylo k dispozici co nejvíce pozorování). Pro volbu prahu je možné využít vizuální techniky - základní jsou dva grafy:

- tzv. mean residual life plot - graf střední hodnoty nadprahových hodnot $E(Y) = (\sigma + \xi u)/(1 - \xi)$ vůči prahu. Střední hodnota nadprahových hodnot by měla růst lineárně.
- graf stability parametru ξ v závislosti na prahu - po dosažení optimální hodnoty prahu by se parametr již neměl příliš měnit.

Oba grafy jsou zobrazeny v sekci 9.4.

9.3 Gumbel plot

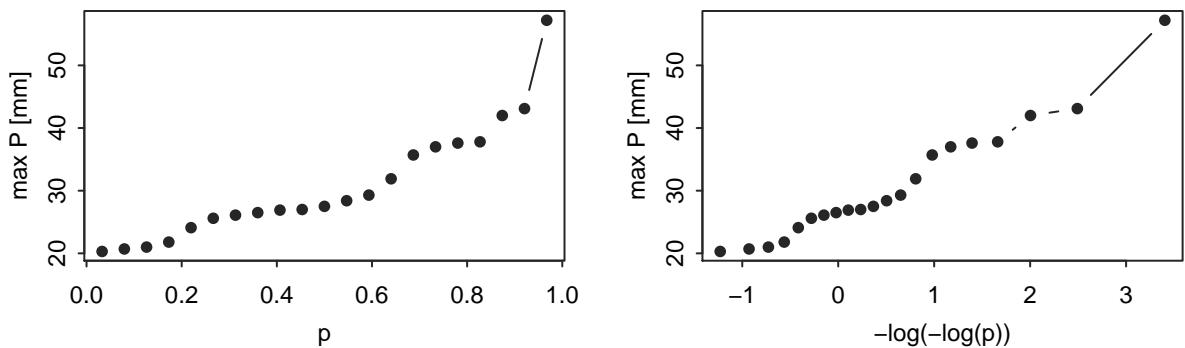
Gumbel plot je grafickým znázorněním transformované distribuční funkce extrémů s pravděpodobností na ose x a maxima na ose y, přičemž pravděpodobnosti (např. p) jsou transformované pomocí vztahu $-\log(-\log(p))$. Tato transformace zajišťuje lepší zobrazení extrémů s vysokou dobou opakování. Gumbelův graf navíc umožňuje diagnostikovat, ke kterému extrémálnímu rozdělení data náleží - data z Gumbelova rozdělení se zobrazí na přímce, z Fréchetova rozdělení jako konvexní funkce a z obráceného Weibullova rozdělení jako konkávní křivka (viz následující graf). Postup pro tvorbu Gumbelova grafu obsahuje následující cvičení.



ÚKOL 9.2 Sestrojte Gumbelův graf pro bloková maxima srážek z příkladu 9.1. Jaké rozdělení mají tato maxima? Použijte následující postup:

- 1 vložte roky a maxima do data.tablu
- 2 seřaďte data.table maxim od nejmenšího k nejvyššímu, použijte funkci `setkey`
- 3 přiřaďte jednotlivým ročním maximům pravděpodobnosti p , přidejte je do data.tablu
- 4 vytvořte v data.tablu novou proměnnou, která bude obsahovat transformované pravděpodobnosti (tj. $-\log(-\log(p))$)
- 5 vykreslete transformované pravděpodobnosti na osu x a maxima na osu y
- 6 zhodnoťte, jaké rozdělení mají maxima
- 7 pro porovnání vykreslete i graf, kde na ose x budou netransformované pravděpodobnosti

```
> MX = data.table(ROK = rok, P = MX)
> setkey(MX, P)
> MX[, p := (rank(P) - .3)/(length(P) + .4)]
> MX[, lp := -log(-log(p))]
> plot(MX[, p], MX[, P], type = 'b', ylab = 'max P [mm]', xlab = 'p', pch=20)
> plot(MX[, lp], MX[, P], type = 'b', xlab = '-log(-log(p))', ylab ='max P [mm]', pch=20)
```



Obr. 9.8: Empirická distribuční funkce (vlevo) a Gumbelův graf (vpravo).

□

9.4 Extrémy v R

Jedním z mnoha balíků umožňující modelování extrémů v Rku je balík evd. Tento balík zprostředkovává rodiny funkcí popisující rozdělení extrémů, tedy dgev, pgev, qgev, rgev a dgpd, pgpd, qgpd a rgpd – hustotu, distribuční a kvantilovou funkci a generátor náhodných čísel z GEV a GPA rozdělení. Mimo to balík obsahuje funkce pro odhad parametrů GEV a GPA modelu - fgev a fpot. Tedy

```
> MX[, fgev(P)]
> dat[, fpot(P, threshold=20)]
```

Výstupem je proměnná typu seznam, udávající informace o statistickém modelu a odhadnutných parametrech. Nejdůležitější položkou je proměnná estimate, udávající optimalizované parametry.

Volba prahu GPA modelu je zpravidla subjektivní záležitost a rozmezí, v němž je možné práh volit, je relativně široké. Pragmatický přístup je zvolit co nejnižší přijatelnou hodnotu (aby počet nadprahových hodnot byl co nejvyšší). Zároveň platí, že pokud má být GPA model výhodný oproti GEV modelu, musí být počet nadprahových hodnot cca 1.2krát vyšší než počet blokových maxim. Základními grafickými prostředky jsou mean residual life plot a graf stability shape parametru:

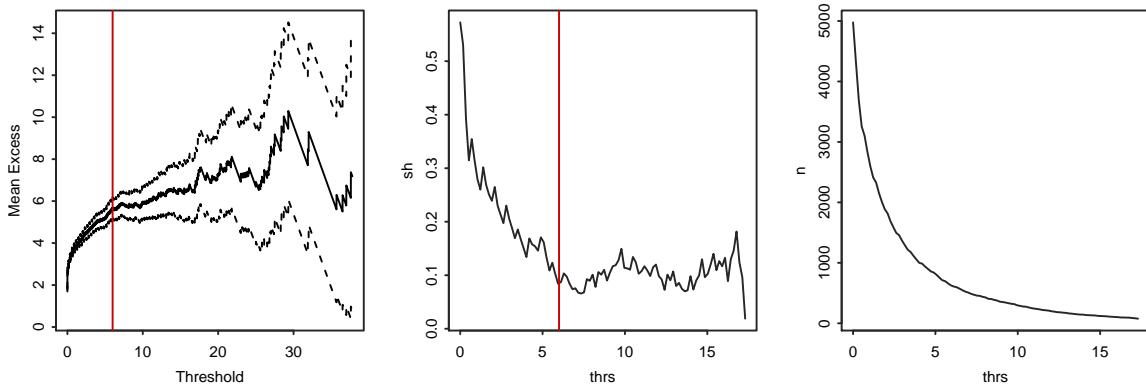
```
> dat
      DTM      P
1: 1930-01-01 0.8
2: 1930-01-02 8.0
3: 1930-01-03 0.0
4: 1930-01-04 0.0
5: 1930-01-05 0.0
---
7666: 1950-12-27 3.7
7667: 1950-12-28 0.3
7668: 1950-12-29 0.1
7669: 1950-12-30 0.0
7670: 1950-12-31 0.0

> sh = c()
> n = c()
> thrs = seq(quantile(dat$P, .1), quantile(dat$P, .99), len=100)
>
```

```

> for (i in 1:length(thrs)){
+   fit = dat[, fpot(P, thre = thrs[i], std.err=FALSE)]
+   sh[i] = fit$estimate['shape']
+   n[i] = fit$nhigh
+ }
>
> par(mfrow=c(1, 3))
> mrlplot(dat$P, main='')
> abline(v=6, col='red3')
>
> plot(thrs, sh, type='l')
> abline(v=6, col='red3')
> plot(thrs, n, type='l')

```



Obr. 9.9: Mean residual life plot (vlevo), shape parametr stability plot (uprostřed) a počet událostí nad zvoleným prahem (vpravo).

Při použití statistického modelu (obecně jakéhokoliv modelu) je nutné prověřit, jak věrně model odpovídá měřeným datům. To je možné buď na základě různých indexů, nebo vizuálně. Pro statistické modelování extrémů je obvyklé prověřovat model vizuální pomocí tzv. quantile-quantile (QQ) grafů a pomocí grafů distribuční funkce. QQ graf vykresluje proti sobě empirické (pozorování) a teoretické (odhadnuté GEV) kvantily extrémů. V Rku (prostřednictvím balíku evd) je možno jej zobrazit pomocí funkce qq. Argumentem je objekt (GEV nebo GPA model) s odhadnutými parametry (pomocí funkcí fgev či fpot), případně další standardní argumenty funkce plot. Distribuční funkce lze zobrazit pomocí funkce rl - tj. return level plot - graf doby opakování a velikosti veličiny.

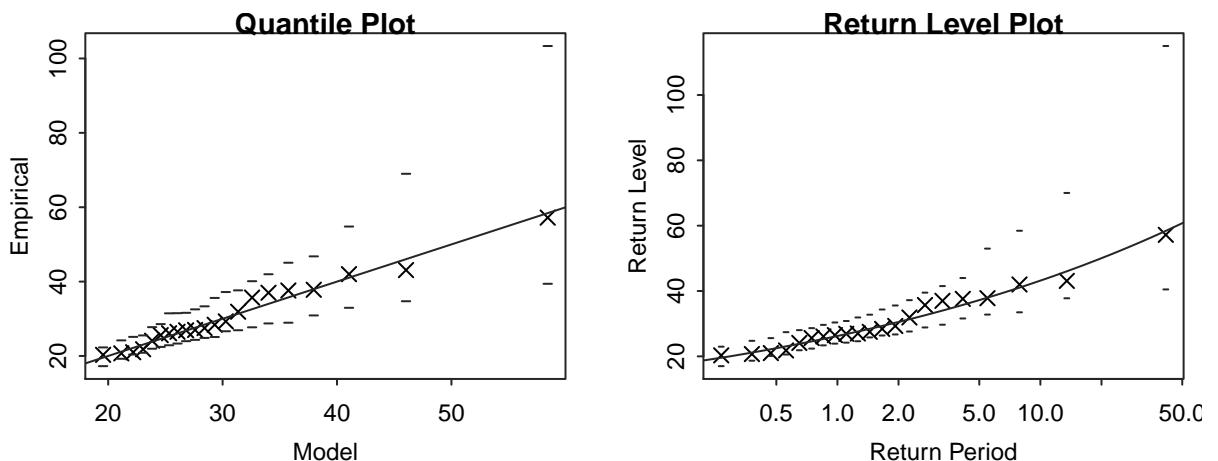
```

> fit = MX[, fgev(P)]
> qq(fit)
> rl(fit)

```

Tab. 9.1: Nejpoužívanější rozdělení pro modelování extrémů.

exp	Exponential
gam	Gamma
gev	Generalized extreme-value
glo	Generalized logistic
gpa	Generalized Pareto
gno	Generalized normal
gum	Gumbel (extreme-value type I)
kap	Kappa
ln3	Lognormal
nor	Normal
pe3	Pearson type III
wei	Weibull



Obr. 9.10: QQ graf pro GEV model (vlevo) a Return level plot (vpravo).

Funkce `fgev` a `fpot` používají k odhadu parametrů rozdělení metodu maximální věrohodnosti. Nicméně je možné i několik jiných způsobů, jeden z nich je metoda L-momentů.

Při modelování extrémů se v praxi používá řada jiných rozdělení než GEV a GPA (viz Tabulka 9.1). Tato rozdělení jsou v R implementovaná v balíku `lmom` nebo `lmomco`. Tyto balíky umožňují odhad parametrů uvedených (a dalších) rozdělení metodou L-momentů. L-momenty jsou obdobou klasických momentů rozdělení, ale jsou založeny na pořadí. Zjednodušeně můžeme přirovnat rozdíl mezi klasickými momenty a L-momenty k rozdílu mezi průměrem a mediánem. Obecně jsou výběrové L-momenty definovány jako

$$\lambda_r = r^{-1} \binom{n}{r}^{-1} \sum_{x_1 < \dots < x_j < \dots < x_r} (-1)^{r-j} \binom{r-1}{j} x_j. \quad (9.10)$$

kde r je řád momentu, n je velikost výběru a $\binom{\cdot}{\cdot}$ je binomický koeficient, vyčíslený jako $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. První čtyři L-momenty se nazývají průměr, L-rozptyl, L-šikmost a L-špičatost a lze je vypočítat pomocí

$$\ell_1 = \binom{n}{1}^{-1} \sum_{i=1}^n x_{(i)} \quad (9.11)$$

$$\ell_2 = \frac{1}{2} \binom{n}{2}^{-1} \sum_{i=1}^n \left\{ \binom{i-1}{1} - \binom{n-i}{1} \right\} x_{(i)} \quad (9.12)$$

$$\ell_3 = \frac{1}{3} \binom{n}{3}^{-1} \sum_{i=1}^n \left\{ \binom{i-1}{2} - 2 \binom{i-1}{1} \binom{n-i}{1} + \binom{n-i}{2} \right\} x_{(i)} \quad (9.13)$$

$$\ell_4 = \frac{1}{4} \binom{n}{4}^{-1} \sum_{i=1}^n \left\{ \binom{i-1}{3} - 3 \binom{i-1}{2} \binom{n-i}{1} + 3 \binom{i-1}{1} \binom{n-i}{2} - \binom{n-i}{3} \right\} x_{(i)} \quad (9.14)$$

V praxi se často používají i tzv. L-moment poměry - tj. standardizované L-momenty

$$\tau_r = \lambda_r / \lambda_2, \quad r = 3, 4, \dots \quad (9.15)$$

zejména pro $r = 3$ a $r = 4$.

V R můžeme L-momenty výběru zjistit pomocí funkce `samlmu`

```
> require(lmom)
> x = MX[, P]
> samlmu(x)

  l_1      l_2      t_3      t_4
30.8333  4.9971  0.2572  0.1544
```

Standardně jsou ve výstupu zahrnutы poměry (t_3 , t_4), skutečné L-momenty do libovolného řádu lze získat pomocí

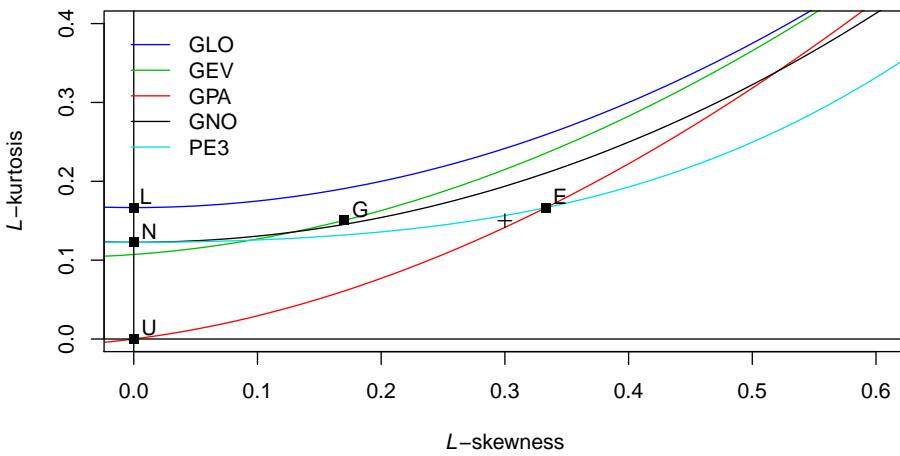
```
> samlmu(x, ratios = FALSE, nmom=7)

  l_1      l_2      l_3      l_4      l_5      l_6      l_7
30.8333  4.9971  1.2852  0.7715  0.3780  0.3935  1.0165
```

L-momenty poskytují robustní odhad parametrů rozdělení, zejména v případě malých výběrů. Odhad parametrů rozdělení je možno získat pomocí funkce typu `parxxx` (balík `lmomco`) nebo `pelxxx` (balík `lmom`), kde za `xxx` je dosazen třípísmený kód z Tabulky 9.1 (balík `lmomco` zná mnoho dalších rozdělení). Vhodné rozdělení se zpravidla v kontextu L-momentů a analýzy extrémů vybírá pomocí tzv. L-moment ratio diagramů, zvýrazňující teoretické hodnoty pro vybraná rozdělení a bod/body odvozené na základě výběru. Tříparametrická rozdělení tvoří v diagramu křivky, dvouparametrická rozdělení tvoří body.

Například pokud by byla L-šíkmost 0.3 a L-špičatost 0.15 L-moment ratio diagram bude vypadat následovně:

```
> lmrd(c(t_3=.3, t_4=.15))
```



a jako vhodné rozdělení bychom zvolili GPA, PE3 nebo EXP.

10 Hydroklimatické indexy

10.1 Klimatické indexy

Klimatické indexy jsou zde definovány jako vypočtené hodnoty, které lze použít k popisu stavu a změn klimatického systému. Klima je dlouhodobý stav počasí, podmíněný energetickou bilancí, cirkulací atmosféry a charakterem aktivního povrchu. Změny týkající se klimatu jsou mnohem pomalejší než změny počasí, které se může měnit ze dne na den. První klasické klimatické indexy atmosféry byly definovány přibližně před sto lety, například Severoatlantická oscilace (NAO).

Každý klimatický index je založen na určitých parametrech a popisuje pouze některé aspekty klimatu. Pro každý index je definována rovnice, do které jsou vstupem klimatologické veličiny, které jsou pozorovatelné a jedná se především o atmosférické veličiny, jako je tlak vzduchu, teplota vzduchu, srážkové úhrny a sluneční záření, ale také neatmosférické veličiny, jako je teplota oceánů či ledové pokrývky. Pro každou veličinu je možné vypočítat extrémní hodnoty, lineární trendy, standardní odchylky z dlouholetých řad atd. Tyto indexy potom řadíme mezi indexy *jednoduché*. Kromě toho existují speciální klimatické indexy využívající více proměnných, tyto indexy potom nazýváme jako *složené*. Pro výpočet některých indexů se používá tzv. referenční období, které by mělo obsahovat alespoň 30 let údajů (doporučení WMO). V současné době je nejpoužívanější období 1961–1990. Průměrné hodnoty jednotlivých parametrů z tohoto období se používají pro další výpočty. Při srovnávání výsledků je důležité, aby porovnávané hodnoty byly porovnávány se stejným obdobím. Mezi nejznámější patří sady indexů pro teplotu vzduchu a srážkové úhrny CLIVAR a STARDEX, jejichž přehled je uveden níže.

Klimatické indexy CLIVAR vyhodnocující teplotu vzduchu:

- *FD, Počet mrazových dní* - Roční počet dnů, kdy TN (denní minimální teplota) $\leq 0^{\circ}\text{C}$.
- *SU, Počet letních dní* - Roční počet dnů, kdy TX (denní maximální teplota) $\geq 25^{\circ}\text{C}$.
- *ID, Počet mrznuocích dní* - Roční počet dnů, kdy TX (denní maximální teplota) $\leq 0^{\circ}\text{C}$.
- *TR, Počet tropických nocí* - Roční počet dnů, kdy TN (denní minimální teplota) $\geq 25^{\circ}\text{C}$.
- *GSL Délka vegetačního období* - Délka vegetačního období začíná, pokud je 6 dní (po sobě jdoucích) s průměrnou teplotou větší než 5°C , a končí, pokud je 6 dní s teplotou nižší než 5°C .
- *TN_x , Maximální denní teplota v daném měsíci* - Maximální naměřená teplota v daném měsíci.
- *TN_x , Maximální minimální denní teplota v daném měsíci* - Maximální minimální naměřená teplota v daném měsíci.
- *TX_n , Minimální denní teplota v daném měsíci* - Minimální naměřená teplota v daném měsíci.
- *$TN10p$, Procento dnů, kdy $TN < 10\%$* - Počet dnů v roce, jejichž průměrná denní teplota je menší než percentil 10% vypočítaný z minimálních teplot.
- *$TX10p$, Procento dnů, kdy $TX < 10\%$* - Počet dnů v roce, jejichž průměrná denní teplota je menší než percentil 10% vypočítaný z maxilmálních teplot.
- *$TN90p$, Procento dnů, kdy $TN > 90\%$* - Počet dnů v roce, jejichž průměrná denní teplota je menší než percentil 90% vypočítaný z minimálních teplot.
- *$TX90p$, Procento dnů, kdy $TX > 90\%$* - Počet dnů v roce, jejichž průměrná denní teplota je

menší než percentil 90% vypočítaný z maximálních teplot.

- *WSDI, Warm SPEEL index* - Roční počet dnů s nejméně šesti po sobě jdoucími dny, kdy $TX > 90\%$ percentil.
- *CSDI, Cold SPEEL index* - Roční počet dnů s nejméně šesti po sobě jdoucími dny, kdy $TN < 10\%$ percentil.
- *DTR, Denní teplotní rozsah* - Měsíční průměrný rozdíl mezi TX a TN .

Klimatické indexy CLIVAR vyhodnocující srážkové úhrny:

- *Rx1day, Měsíční maximum 1-denní srážky* - Maximální 1-denní hodnota srážkového úhrnu pro daný měsíc.
- *Rx5day, Měsíční maximum po sobě 5-denní srážky* - Maximální 5-denní hodnoty srážkového úhrnu pro daný měsíc.
- *R10mm, Roční počet dnů, kdy srážka 10 mm* - Počet dnů v roce, kdy srážkový úhrn je větší než 10 mm.
- *R20mm, Roční počet dnů, kdy srážka 20 mm* - Počet dnů v roce, kdy srážkový úhrn je větší než 20 mm.
- *CDD, Maximální délka období sucha* - Maximální počet po sobě jdoucích dnů v roce, kdy srážkový úhrn je menší než 5 mm.
- *CWD, Maximální délka mokra* - Maximální počet po sobě jdoucích dnů v roce, kdy srážkový úhrn je větší než 5 mm

Klimatické indexy STARDEX:

- 90ti procentní kvantil denních srážkových úhrnů
- maximální 5-denní srážkový úhrn v daném roce
- průměrná denní intenzita
- maximální počet po sobě jdoucích suchých dní
- počet po sobě jdoucích dnů, kdy srážkový úhrn je větší než 90 % a počet těchto událostí v daném roce
- 90 procentní kvantil denních teplot vzduchu
- 10 procentní kvantil denních teplot vzduchu
- počet dní přesahující 90 % kvantil v daném roce
- počet dní v roce, kdy průměrná denní teplota je menší než 0°C

Důležité jsou samozřejmě také indexy popisující změny tlaku, toto téma je však nad rámec těchto skript.

ÚKOL 10.1 Vykreslete graf minimálních měsíčních teplot.

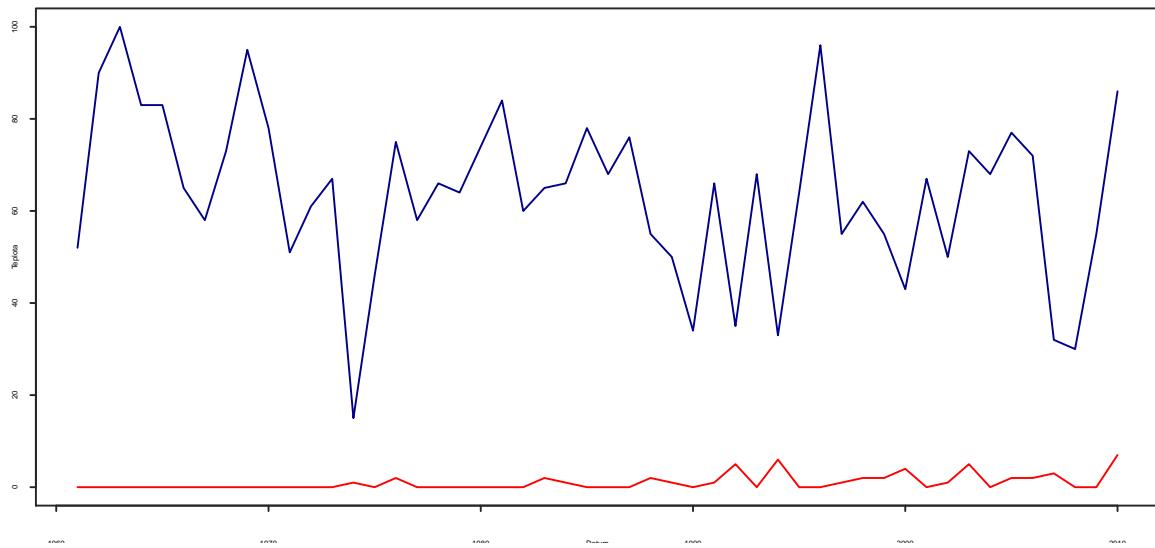
```
> require(data.table)
> dta = data.table(readRDS("../P_T.rds"))
> TNX = dta[, min(TEMP), by = list(year(DTM), month(DTM))]
> plot(seq(as.Date("1961/1/1"), by = "month", length.out = length(TNX$V1)), TNX$V1,
+       ylab = "Teplota", xlab = "Datum", type = "l")
```



Obr. 10.1: Graf minimálních měsíčních teplot

ÚKOL 10.2 Vypočítejte počet letních a mrznoucích dní, hodnoty vykreslete do grafu.

```
> par(par_plot)
> LD = dta[, sum(TEMP > 25), by = year(DTM)]
> MD = dta[, sum(TEMP < 0), by = year(DTM)]
> plot(seq(as.Date("1961/1/1"), by = "year", length.out = length(MD$V1)), MD$V1,
+       ylab = "Teplota", xlab = "Datum", type = "l", ylim = c(0, 100), col = "dark blue")
> lines(seq(as.Date("1961/1/1"), by = "year", length.out = length(MD$V1)), LD$V1,
+       col = "red")
```



Obr. 10.2: Graf mrznoucích a letních dní

10.2 Indexy pro hodnocení nedostatku vody

Sucho je jednou z nejzávažnějších pohrom souvisejících s počasím, zároveň je to málo prozkoumaná přírodní katastrofa. Také se od ostatních liší v mnoha směrech. Je to pomalu se projevující stav, který ani na začátku, ani na konci nemusí být jasně definován. Sucho musí být vnímáno jako přirozená součást klimatu za všech klimatických režimů. Jeho prostorové a časové charakteristiky se značně liší region od regionu. Vyskytuje se ve vysokých i nízkých srážkových oblastech. Frekvence sucha je často spojena se specifickými klimatickými regiony (např. je běžné v africkém Sahelu nebo Austrálii, v západní a střední Evropě se naopak objevuje jen zřídka) a nejzávažnější lidské důsledky sucha se často vyskytují v suchých a polosuchých oblastech, kde je dostupnost vody nízká již za normálních podmínek. Sucho by však nemělo být zaměňováno s vyprahlostí, což je dlouhodobý rys suchého klimatu, nebo s nedostatkem vody, což odpovídá podmínkám dlouhodobé nerovnováhy mezi dostupnými vodními zdroji a požadavky na ně. Také klesající trend v dostupnosti vody není sucho, ale tzv. vysojení, aridifikace nebo desertifikace. Další důležitou definicí je rozdíl mezi nízkým průtokem a suchem. Nízké průtoky se v oblastech s jasnon sezoností každoročně opakují (Blinka, 2004; Peters, 2003; Tallaksen a van Lanen, 2004; Bratršovská, 2013).

Sucho a nedostatek vody jsou pojmy, které je třeba od sebe rozlišovat.

Nedostatek vody je zde definován jako situace, kdy vodní zdroj není dostatečný pro uspokojení dlouhodobých průměrných požadavků na vodu.

Sucho představuje dočasný pokles průměrné dostupnosti vody a je považováno za přirozený jev.

10.3 Sucho

Sucho hodnotíme z prostorového a časového hlediska, určuje se také jeho intenzita. Kromě času se na charakteru sucha podílejí také další faktory jako je teplota vzduchu, rychlosť větru, relativní vlhkost vzduchu atd. Pro stanovení existence sucha a jeho intenzity existuje mnoho objektivních metod, do kterých jsou vstupem četné meteorologické veličiny (srážkový úhrn, teplota vzduchu, reálná a potenciální evapotranspirace, půdní vláha, povrchový odtok, infiltrace vody do hlubších vrstev, zásoba vody ve sněhu, v řekách a nádržích). Vstupem budou jednak časové řady pozorované (teplota vzduchu, průtoky, ...) a modelované modelem (potenciální a reálná evapotranspirace, infiltrace, ...).

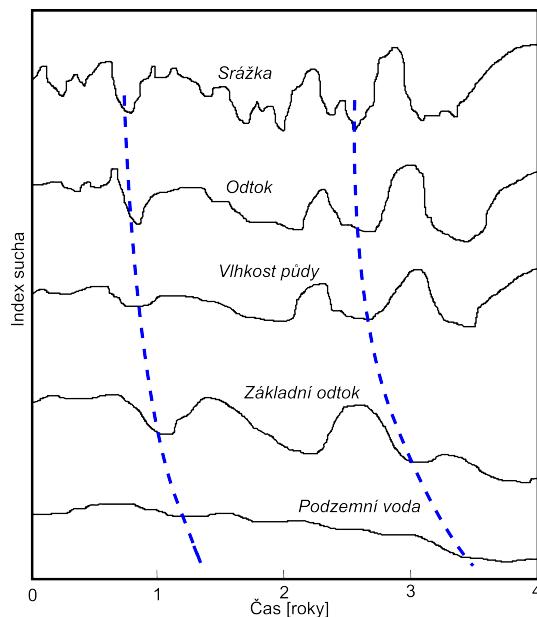
Suchá období mívají různou intenzitu i různou délku trvání od krátkodobých až po několikaměsíční. Následky mohou být také závislé na ročním období, v němž se sucho vyskytne.

10.3.1 Propagace sucha

Propagace sucha (viz obrázek 10.3) je proces, ve kterém deficit srážkových úhrnů následně vede ke snížení povrchového odtoku, abnormálnímu deficitu půdní vlhkosti, podzemní vody a/nebo průtoků (Wanders et al., 2010).

Sucha jsou často klasifikována do čtyř kategorií, které jsou založeny na různých částech hydrologického cyklu a dělíme je na:

- 1 Meteorologické sucho



Obr. 10.3: Propagace sucha

- 2 Agronomické sucho
- 3 Hydrologické sucho
- 4 Socioekonomicke sucho

Meteorologické sucho

Primární příčinou meteorologického sucha je deficit srážek v určitém časovém intervalu, jenž může být prohlouben spolupůsobením ostatních meteorologických prvků, zejména vyššími teplotami vzduchu, intenzivnějším prouděním vzduchu či jeho nízkou relativní vlhkostí. Ve své „nejmírnější“ podobě nemusí působit žádné větší škody, obvykle se hodnotí na základě odchylky srážek od normálu za určité časové období. Vyjadřuje jednu z primárních příčin sucha, jakožto záporná odchylka srážek od normálu za určité časové období podmiňuje výskyt sucha zemědělského, hydrologického i socioekonomického. Kromě množství a intenzity spadlých srážek vztažených k srážkovým normálům pro danou lokalitu a roční dobu stanovili mnozí autoři různé definice meteorologického sucha v závislosti na dalších meteorologických prvcích (především na výparu, teplotě vzduchu, rychlosti větru, vlhkosti vzduchu aj.), pomocí klimatologických indexů. Meteorologické sucho je někdy nesprávně nazýváno suchem atmosférickým.

Agronomické sucho

Jako zemědělské sucho označujeme období, kdy panuje dlouhodobější nedostatek vody v půdě a její dostupnost rostlinám se stává limitem jejich normálního růstu a vývoje. Zemědělské sucho je vyvolané předchozím nebo nadále trvajícím výskytem meteorologického sucha, často v kombinaci s vysokou ztrátou evapotranspirací. Z dalších vlivů mají značný význam vlastnosti půdy, úroveň zemědělské techniky, která se v dané oblasti používá, a celá řada dalších faktorů. Definice zemědělského sucha je

obšírně diskutovaným problémem, který předpokládá podrobné znalosti z hydropedologie, rostlinné fyziologie, ekonomiky a příbuzných oborů.

Hydrologické sucho

Hydrologické sucho je definováno pro povrchové toky určitým počtem za sebou jdoucích dní, týdnů, měsíců i roků s výskytem nízkých průtoků vzhledem k měsíčním či ročním normálovým hodnotám. Hydrologické sucho se vyskytuje zpravidla ke konci déle trvajícího období sucha, ve kterém nepadaly kapalné ani smíšené srážky. Obdobných kritérií je možno použít i pro stavy hladin podzemních vod a výdatnosti pramenů. Výskyt hydrologického sucha předznamenává nejvážnější škody způsobené suchem. Tento druh sucha se často vyskytuje i v době, kdy již meteorologické sucho dávno odeznělo. Naopak při výskytu meteorologického sucha se ještě nemusí jednat o sucho hydrologické. Studium hydrologického sucha znamená studium bezvodých (resp. málovodých) období a jejich parametrů.

Historická období hydrologického sucha lze charakterizovat různými veličinami: dosaženými minimy průtoků, dosaženými minimy průtoků z klouzavých průměrů (např. 7 až 30-denními), nedostatkovými objemy a trváním (objemy chybějícími pod určitou mezí průtoku a trvání průtoků pod určitou mezí) aj. Dalším kritériem výskytu sucha může být významný pokles hladiny podzemních vod. Historická sucha zpravidla postihují území celé České republiky, o míře extremity v dané oblasti potom rozhodují zejména místní dlouhodobější srážkové poměry. Období sucha navíc většinou doprovází nadprůměrné teplotní poměry, které dále zhoršují vodní bilanci.

Socioekonomicke sucho

Definice socioekonomickeho sucha spojuje sucho s ekonomickeou teorií. O socioekonomickeém suchu hovoříme tehdy, je-li intenzita či délka suché periody natolik závažná, že má přímý vliv na obyvatelstvo (snížení dostupnosti zdrojů pitné vody) a ekonomiku země (ohrožení zemědělské výroby v masivním měřítku, narušení výrobně obchodních vztahů). Definice socio-ekonomickeho sucha může částečně překrývat definici jak zemědělského, tak i hydrologického sucha.

10.4 Vybrané indexy pro hodnocení sucha

10.4.1 SPI

Tento ukazatel (SPI-Standard Precipitation Index) byl zaveden v roce 1993 (McKee et al., 1993) k monitorování a určení suchých období. Na rozdíl od jiných indexů má několik výhod: pro výpočet jsou nutná pouze srážková data, výpočet je relativně snadný (zavádí se jen dva další parametry), a standardizovaný charakter. Posledně zmíněná věc však může být zároveň nevýhodou. Extrémně suchá období budou klasifikována se stejnou frekvencí jako extrémně vlhká období na různých lokalitách. Proto se doporučuje použít jako doplňující informace k jiným ukazatelům (Lloyd a Saunders, 2002). V prostředí R jsou pro výpočet indexu SPI určeny balíky `spi` a `SPEI`.

Jedná se vlastně o transformaci srážkových časových řad na normální rozdělení. Měsíční (nebo jiný časový interval) jsou approximovány pravděpodobnostním rozdělením (nejčastěji se používá gama

rozdělení, ale v některých případech může být vhodnější Poissonovo nebo Log-normální). České lokality se zpravidla approximují gama rozdělením s obdobným nebo lepším výsledkem než log-normální rozdělení (Obrázek 10.4).

gama rozdělení

Je dané následující funkci, proměnná x odpovídá srážkovým úhrnům, parametry α a β určují tvar křivky.

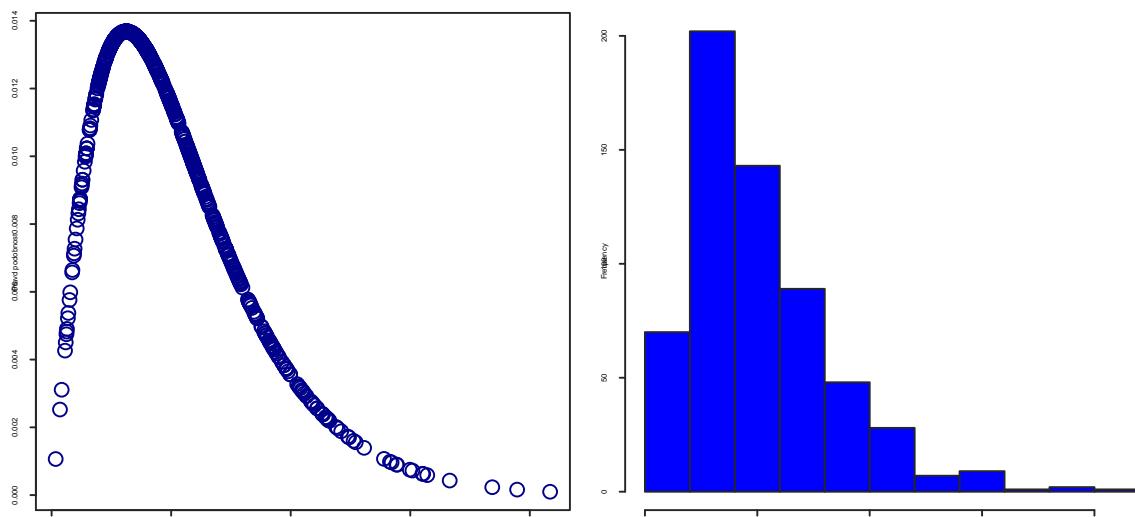
$$g(x) = \frac{1}{\beta^\alpha(\alpha)} x^{\alpha-1} e^{-x/\beta} \quad \text{pro } x > 0 \quad (10.1)$$

ÚKOL 10.3 Vykreslete graf a histogram srážkových úhrnů. Gama rozdělení má parametry: $shape=2.3$, $scale=24$.

```
> srazka = dta[, sum(PRECIP), by = list(month(DTM), year(DTM))]
> head(srazka)

  month year    V1
1:     1 1961 16.2
2:     2 1961 42.9
3:     3 1961 48.0
4:     4 1961 48.9
5:     5 1961 89.3
6:     6 1961 78.2

> pravdepodobnost = dgamma(srazka$V1, shape = 2.3, scale = 24)
> plot(srazka$V1, pravdepodobnost, col = "dark blue", xlab = "Srážkový úhrn",
+       ylab = "Pravděpodobnost")
> hist(srazka$V1, col = "blue", xlab = "Srážkový úhrn", main = "")
```



Obr. 10.4: Příklad gama rozdělení, $\alpha = 2.3$ a $\beta = 24$

Tyto parametry je třeba určit pro každou oblast a časový interval, obvykle se používá následujících vztahů:

$$A = \ln(\bar{x}) - \frac{\sum \ln(x)}{n} \quad (10.2)$$

Tab. 10.1: Klasifikace SPI

Hodnota indexu	Charakter období
$\zeta = 2$	Extrémně vlhký
1,5 až 2	Velmi vlhký
1 až 1,49	Mírně vlhký
0 až 0,99	Slabě vlhký
0 až -0,99	Slabě suchý
-1 až -1,49	Mírně suchý
-1,5 až -1,99	Velmi suchý
$\zeta = -2$	Extrémně suchý

$$\hat{\alpha} = \frac{1}{4A} \left(1 + \sqrt{1 + \frac{4A}{3}} \right) \quad (10.3)$$

$$\hat{\beta} = \frac{\bar{x}}{\hat{\alpha}} \quad (10.4)$$

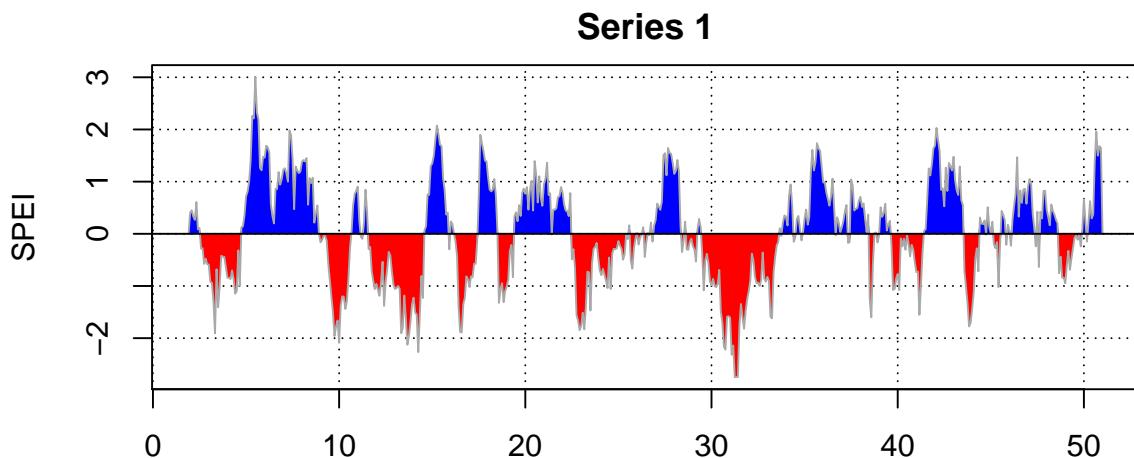
Pro daný srážkový úhrn se pak vypočte distribuční funkce (kumulativní pravděpodobnost) k již parametrizovanému gama rozdělení:

$$G(x) = \int_0^x g(x) dx = \frac{1}{\hat{\beta}^{\hat{\alpha}}(\hat{\alpha})} \int_0^x x^{\hat{\alpha}} e^{-x/\hat{\beta}} dx \quad (10.5)$$

a výsledná hodnota se transformuje zpátky na normální rozdělení.

ÚKOL 10.4 Vypočítejte index sucha SPI pro 12 měsíců pomocí balíku SPEI a vykreslete do grafu

```
> library("SPEI")
> spi_12 = spi(srazka$V1, 12)
> plot(spi_12, main = "")
```



Obr. 10.5: Graf SPI indexu pro 12 měsíců

10.4.2 PDSI - Palmer Drought Severity Index

Byl zaveden v šedesátých letech v USA (Palmer, 1965). Na rozdíl od předchozího klimatického ukazatele bilancuje nejen srážky v dané oblasti, ale zároveň i zásobu vody v půdním horizontu a výpar. Umožnuje tak kvantifikovat a porovnávat sucho v oblastech s odlišnými pedologickými a klimatickými poměry. Rekurzivní formule pro tento index vychází z odchylky od klimatického normálu pro daný časový interval a z předešlého výpočetního období. Metoda výpočtu je tím pádem nastavena tak, aby jeden suchý měsíc (s nízkým srážkovým úhrnem) v dlouhodobě vlhkém období neměl zásadní vliv na hodnotu indexu.

Vstupní data

Opět se používají srážkové řady, dále hodnoty aktuální výpar a potenciální evapotranspirace a tzv. využitelná vodní kapacita.

Metoda výpočtu

Předpoklady / Zjednodušení.

Původní Palmerova metodologie používá dvě vrstvy půdního horizontu, spodní horizont se může nasýtit vodou, až když je nasycen svrchní, obdobné platí pro výpar. Vzhledem k odlišným pedologickým poměrům a chybějící datům ohledně stratifikace půdního horizontu pro jednotlivé oblasti se jeví jako vhodnější aplikovat modifikovaný PDSI (mPDSI), jenž uvažuje pouze jednu půdní vrstvu. (Kingste a Chelliah, 2006) ukázali použitelnost tohoto modifikovaného indexu pro USA s podobnými výsledky jako při aplikaci PDSI, navíc mPDSI vychází ze stejných principů jako je výpočet PDSI. Odhad výšky svrchního půdní vrstvy by byl další subjektivní vstup do výpočtu, použitím jednovrstvého půdního horizontu se tak vyhneme možnému zdroji chyby. Výpočet PDSI předpokládá, že k povrchovému odtoku dochází pouze tehdy, když je nasycena půdní vrstva a srážky převyšují výpar. Nezohledňuje se tak případná zásoba vody ve sněhové vrstvě.

Postup výpočtu

Nejdříve se pro každý řešený časový interval určí odchylka srážek od „klimatického normálu“ (v

anglickém jazyce se používá zkratka CAFEC – climatically appropriate for existing conditions):

$$d_i = P_i - \hat{P}_i \quad (10.6)$$

$$\hat{P}_i = \alpha_j PE_j + \beta_j PR_j + \gamma_j PRO_j - \delta_j PL_j \quad (10.7)$$

Použité koeficienty pak odpovídají poměrům aktuálních hodnot příslušných veličin k jejich potenciálním hodnotám:

$$\alpha_j = \frac{\overline{ET}_j}{\overline{PE}_j} \quad \beta_j = \frac{\overline{R}_j}{\overline{PR}_j} \quad \gamma_j = \frac{\overline{RO}_j}{\overline{PRO}_j} \quad \delta_j = \frac{\overline{J}_j}{\overline{PL}_j} \quad (10.8)$$

kde: ET - aktuální evapotranspirace [mm],
 PET - potenciální evapotranspirace [mm],
 PR - potenciální doplnění, množství vláhy potřebné k doplnění půdního profilu na využitelnou vodní kapacitu (VVK) [mm],
 R - aktuální navýšení půdní vlhkosti [mm],
 RO - odtok, je přebytek srážek proti výparu v případě, že je půdní horizont nasycen [mm],
 PRO - potenciální odtok odpovídá rozdílu mezi potenciálními srážkami (resp. VVK) a potenciálním doplněním, tj. $PRO = \max(P) - PR$ [mm],
 PL - potenciální ztráta odpovídá množství srážek, které lze z půdy odebrat evapotranspirací v případě nulových srážek. tj. $PL = \min(PE, S)$, kde S je zásoba vody v půdě. [mm],
 L - odtok, odpovídá úbytku vláhy v půdě, obdoba doplnění. [mm].

Na základě d se určí odchylky od půdní vlhkosti (index z , který se někdy rovněž používá jako ukazatel sucha):

$$z_i = K_i d_i, \quad (10.9)$$

kde: K se určí dle následujících vztahů:

$$K_j = \left(\frac{17,67}{\sum_{i=1}^{12} K'_j D_j} \right) K'_j, \quad (10.10)$$

$$K'_j = 1,5 \log \left(\frac{PE_j + R_j + RO_j}{P_j + L_j} \right) + 0,5. \quad (10.11)$$

D_j je průměr absolutních odchylek d pro každou hodnotu.

Koeficient PDSI se pak vyjádří jako:

$$PDSI_i = 0,897 PDSI_{i-1} + \frac{1}{3} z_i. \quad (10.12)$$

Hodnoty indexu se pak klasifikují dle následující tabulky (Tab. 10.2):

Tab. 10.2: Klasifikace PDSI

Hodnota indexu	Charakter období
$\zeta = 4$	Extrémně vlhký
3 až 3,99	Velmi vlhký
2 až 2,99	Mírně vlhký
1 až 1,99	Slabě vlhký
1 až -1	Normální stav
-1 až -1,99	Slabě suchý
-2 až -2,99	Mírně suchý
3 až 3,99	Velmi suchý
$\zeta = 4$	Extrémně suchý

10.4.3 Nedostatkové objemy

Jedním z hlavních kritérií pro posouzení hydrologického sucha jsou nedostatkové objemy.

Průtok je popsaný časovou funkcí $Z(t)$. Funkce $Z(t)$ je v čase proměnná v požadavku na vodu. Pro časový integrál $\langle t p_i, t k_i \rangle$, pro který je splněná podmínka $Z(t_i) > Q(t_i)$ definujeme určitý integrál W_i :

$$W_i = \int_{tp_i}^{tk_i} [Z(t_i) - Q(t_i)] dt \quad (10.13)$$

kde: $i = 1, 2, 3, \dots, n, n$ - počet deficitů v řešeném období,

$t p_i$ - čas počátku i -tého deficitu,

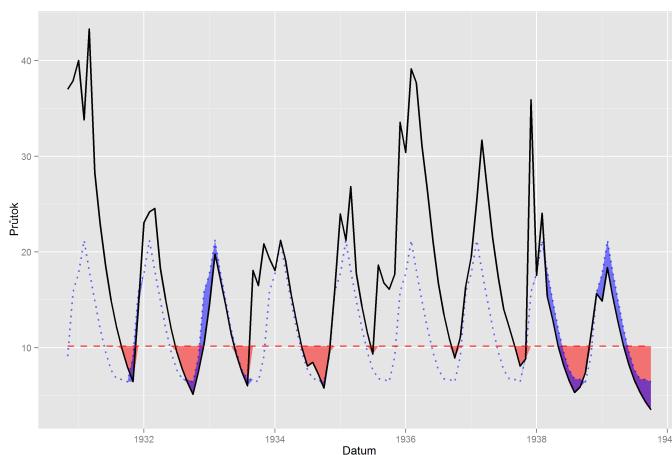
$t k_i$ - čas konce i -tého deficitu.

Rovnice definuje parametry náhodné veličiny S_i i -tého sucha. Je to nedostatkový objem W_i a doba trvání nedostatku vody $T_i = tk_i - tp_i$. Nejčastěji používaným přístupem je volba funkce $Z(t_i) = konst$. V tomto případě se jedná o metodě ořezání průtokové řady na konstantní úroveň průtoku (Bonacci, 1993).

Důležitým faktorem je úroveň hladiny ořezu (threshold level). Bývají to kvantily průměrného ročního průtoku $Q_{80\%}, Q_{90\%}, \dots$ a průtoky např Q_{330}, Q_{355} nebo Q_{364} . V tomto případě jsou jako vstup použity denní průměrné průtoky. Avšak sucho lze posuzovat i v měsíčním časovém kroku. Měsíc je dost dlouhá doba na to, aby se daly postihnout regionální závislosti a specifika. (Bonacci, 1993), poukazuje na to, že sucho není způsobeno jen deficitem srážek, ale změnou rozdělení v průběhu roku. Měsíční krok také umožňuje zahrnutí vlivu užívání vod (v měsíčním kroku jsou ve vodním hospodářství udávány údaje na užívání vod). Měsíční průtokové řady jsou stejně přesné jako denní, pokud jsou vyhotoveny z původní denní řady.

Na obrázku 10.6 je zobrazen příklad pro konstantní a variabilní hladinu ořezu, kde se zvolila hladina $Q_{70\%}$ pro konstantní úroveň a $Q_{70\%m}$ v jednotlivých měsících. Zavedení variabilní hodnoty je zejména důležité pro reflexi ekologických a jiných požadavků na vodní tok, avšak v současné době se v České republice většinou užívá hodnota minimálního zůstatkového průtoku, která se vypočítána z denních průtokových hodnot, a proto není použitelná pro vyhodnocení v měsíčním časovém kroku. Proto se někdy volí průtokové kvantily $Q_{70\%}, Q_{80\%}, Q_{90\%}$ a $Q_{95\%}$, a to pro konstantní i variabilní hladinu ořezu.

10.4.4 Další často užívané indexy



Obr. 10.6: Konstantní nedostatkový objem (červeně) a variabilní (modře), vyplněné polygony jsou nedostatkové objemy pro jednotlivé typy hladiny ořezu

Meteorologické sucho

Percent of Normal (PR)

Výpočet vychází z poměru aktuálních srážek k příslušnému srážkovému normálu, obyčejně třicetiletému. Podstata metody "decilů" vychází z konstrukce hustoty, která se může lišit od hustoty normálního rozdělení. Graf pod frekvenční křivkou se rozdělí na 10 plošně shodných úseků. První decil udává množství srážek, které není překročeno ve více než 10 procentech případů. Pátý decil je medián, který určuje srážku, jež není překročena v 50 % případů. Metoda decilů se vztahuje k meteorologickému suchu a mezi její výhody patří menší náročnost na vstupní údaje.

PR index je jednou z nejjednodušších metod hodnocení srážek pro určité místo, umožňuje rovněž stanovit délku a intenzitu sucha, zejména vzhledem k dlouhodobým poměrům. Nicméně srážky v měsíčním a sezónním měřítku nemají často normální rozdělení, přítom tato metoda vychází právě z předpokladu normálního rozdělení. PR rovněž neumožňuje srovnání mezi různými místy (Blinka, 2004).

Cumulative precipitation anomaly (CPA)

Srážkové anomálie přímo měří nedostatek srážek, a tvoří rozdíl mezi měřenými a dlouhodobým klimatologickým průměrem. Tato anomálie je vlastně primitivním indexem sucha. Není nijak zvlášť informativní, poněvadž význam anomálií závisí na klimatu. Měsíční deficit 1 cm je neporovnatelně více významný pro ekosystém pouště v porovnání například s ekosystémem horského lesa (Keyantash a Dracup, 2002).

Drought area index (DAI)

DAI (1980) je rekurzivní index, v němž postupné hodnoty závisí na předchozí měsíční hodnotě, čili zohledňuje trvání sucha. Tento index je vyjádřen následujícím vztahem (Keyantash a Dracup, 2002; Drlička, 2004):

$$I_k = 0,5I_{k-1} + \frac{R_k - \bar{R}_k}{48,55\sigma_k}, \quad (10.14)$$

kde: I_k - intenzita sucha,

k - číslo měsíce,

R - měsíční úhrn srážek,

\bar{R} - průměrný měsíční úhrn srážek,

σ - směrodatná odchylka R .

Rainfall anomaly index (RAI)

Tento index byl vyvinut van Rooy (1965). Vychází ze srážkových úhrnů a jejich extrémů. Obsahuje klasifikační

postup pro přiřazení veličin k pozitivním a negativním srážkovým anomáliím. Hodnoty indexu jsou následně posuzovány na základě devíti členného klasifikačního systému, od extrémně vlhkých po extrémně suché. Index je dán následujícím vztahem (Keyantash a Dracup, 2002):

$$RAI = 3 \frac{R - \bar{R}}{\bar{E} - \bar{R}}, \quad (10.15)$$

kde: R – měsíční úhrn srážek,

\bar{R} – průměrný měsíční úhrn srážek,

\bar{E} – průměr z deseti nejvyšších, resp. nejnižších hodnot.

Precipitation effectiveness index (P-E, PEI)

P-E index (1931) je měřítkem dostupnosti půdní vlhkosti pro vegetaci, které závisí na množství a distribuci srážek i evaporace. Je definovaný podílem měsíčních srážek R a výparu E (Ehrlich et al., 1970):

$$PE_{index} = 10 \sum_{j=1}^{12} \frac{R_{mj}}{E_{mj}}. \quad (10.16)$$

Agronomické sucho

Crop Moisture Index (CMI)

(Palmer, 1965) vyvinul ke sledování krátkodobých změn ovlivňujících úrodu. Používá se ve Spojených státech amerických. CMI je rozdílem evapotranspiračního deficitu (s ohledem na normální podmínky) a doplnění půdní vlhkosti. Tyto podmínky jsou vypočteny v týdenním časovém kroku za použití PDSI parametrů, které zohledňují průměrné teploty, srážkové úhrny a podmínky půdní vlhkosti z předchozího týdne. CMI může vyhodnotit stávající podmínky pro plodiny, ale může prudce kolísat a je nevhodným nástrojem pro sledování dlouhodobého sucha. Například bouře může krátkodobě zajistit dostatečnou vlhkost pro úrodu, přestože přetrvává rozsáhlé sucho. CMI také začíná a končí každé vegetační období téměř na nule. V důsledku toho se pro posuzování zemědělského sucha lépe hodí Palmerův Z index (Keyantash a Dracup, 2002; Hayes, 2000; Palmer, 1965).

Palmer Moisture Anomaly Index (Z index)

Z-index je vlastně mezírok při výpočtu v PDSI, nicméně je považován za samostatný index. Je anomálií vlhkosti pro aktuální měsíc, ale nebude v úvalu dřívější stav, jako je tomu v případě PDSI. Index může posloužit jako ukazatel zemědělského sucha, protože zaznamená okamžité změny v hodnotách vlhkosti půdy (Keyantash a Dracup, 2002).

Soil moisture anomaly index

Tento index byl vyvinut k charakterizaci sucha na globální úrovni. Metoda se významně opírá o Thornthwaitovu metodu výpočtu půdní vlhkosti a operuje v rámci dvouvrstvého půdního modelu sloužícího ke sledování pohybů vody; v zásadě vyplývá z nepřetržitého hodnocení procenta půdní saturace. Simulace napovídají, že hodnoty Soil Moisture Anomaly Index se pohybují mezi rázným CMI a relativně pomalým PDSI (Keyantash a Dracup, 2002).

Hydrologické sucho

Palmer hydrological drought severity index (PHSI)

PHSI je ve své podstatě hydrologickou obdobou PDSI. Používá stejný model dvouvrstvého půdního profilu sloužícího ke sledování pohybů vody. Rozdíl je v tom, že PHDI má přísnější kritéria pro eliminaci známek sucha či vlhkosti, což vede ke graduálnímu průběhu indexu, a k pomalejšímu návratu k normálnímu stavu než v případě PDSI. PDSI uvažuje konec sucha, když ukazatele vlhkosti začnou nepřerušovaně vzrůstat, až

nakonec převýší vodní deficit. Zatímco PHDI uvažuje konec sucha tehdy, když deficit vlhkosti skutečně zmizí. Toto zpomalení je vhodné pro odhad hydrologického sucha, které je pomaleji se rozvíjejícím fenoménem než meteorologické sucho (Keyantash a Dracup, 2002; Drlička, 2004; Heim, 2002).

The surface water supply index (SWSI)

Autorem SWSI je Shafer Dezman. Tento index vychází z vodních zásob sněhové pokrývky, srážek, průtoku a dalších vodních zásob. Tyto percentily jsou zadávány do na povodí kalibrovaného algoritmu SWSI, který uvažuje určitý příspěvek každého hydrologického komponentu. Tato váha umožňuje porovnávání mezi povodími. Index je používán především na západě Spojených států, protože se jedná o metodu vhodnou pro měření hydrologického sucha v regionech, jako jsou právě Skalnaté hory (kde sníh výrazně přispívá k ročním průtokům) (Keyantash a Dracup, 2002; Drlička, 2004).

Total water deficit

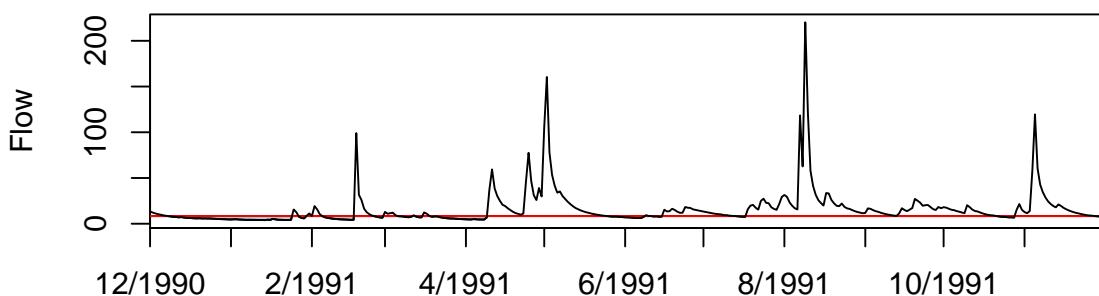
Total water deficit je synonymem pro závažnost sucha S . Tato závažnost je výsledkem trvání D , během kterého jsou průtoky trvale pod úrovní (například hydroklimatickým průměrem), a významnosti M , která je průměrnou odchylkou průtoku od úrovně během období sucha. Poté, co sucho skončí, se Total water deficit znova přiblíží. Trvání a závažnost se v literatuře často objevují jako "run sum", "run length" a "run intensity" (Keyantash a Dracup, 2002; Yevjevich et al., 1967).

Cumulative stramflow

Kumulativní odchylka průtoku od průměrných hodnot může ukázat dlouhodobé tendenze v dostupnosti vody. Příkré poklesy v odchylkách kumulativních průtoků reprezentují sucha (Keyantash a Dracup, 2002).

ÚKOL 10.5 Vypočítejte nedostatkové objemy odtokových výšek. Pro výpočet použijte balík `lfstat`

```
> library("lfstat")
> data(ngaruroro)
> streamdefplot(ngaruroro, year = 1991)
```



Obr. 10.7: Nedostatkové objemy

Literatura

- Blinka, P. (2004) KLIMATOLOGICKÉ HODNOCENÍ SUCHA A SUCHÝCH OBDOBÍ NA ÚZEMÍ ČR V LETECH 1876–2003. *Seminář „Extrémy počasí a podnebí“*, Brno.
- Bonacci, O. (1993) Hydrological identification of drought. *Hydrological Processes*, 7, 249–262.
- Bratršovská, L. (2013) *Vyhodnocení propagace sucha hydrologickým cyklem na povodí Tiché Orlice a Střely - Diplomová práce*. ČZU, Praha.
- Drlička, R. (2004) *Sucha na Moravě a ve Slezku*. Diplomová práce – Masarykovo univerzita.
- Ebert, E. E. (2008) Fuzzy verification of high-resolution gridded forecasts: a review and proposed framework. *Meteorological applications*, 15(1), 51–64.
- Ehrlich, I., Kolb, R., Sloss, D., Corridon, L. (1970) STUDIES OF OFF-ROAD VEHICLES IN THE RIVERINE ENVIRONMENT. VOLUME 3. ASSOCIATED ENVIRONMENTAL FACTORS. Technická zpráva, DTIC Document.
- Gumbel, E. J. (1958) *Statistics of extremes*. Columbia University Press, New York.
- Hayes, M. J. (2000) *Drought indices*. National Drought Mitigation Center, University of Nebraska.
- Heim, R. R. (2002) A review of twentieth-century drought indices used in the United States. *Bulletin of the American Meteorological Society*, 83(8).
- Jarušková, D. (2011) *Pravděpodobnost a matematická statistika*. Skripta ČVUT.
- Jolliffe, I. T., Stephenson, D. B. (2012) *Forecast verification: a practitioner's guide in atmospheric science*. John Wiley & Sons.
- Keyantash, J., Dracup, J. A. (2002) The quantification of drought: An evaluation of drought indices. *Bulletin of the American Meteorological Society*, 83(8).
- Kingste, C. M., Chelliah, M. (2006) *The Modified Palmer Drought Severity index based on the NCEP North American regional analysis*. NOAA/NWS/NCEP Klimatology center.
- Lloyd, H. B., Saunders, M. A. (2002) A drought climatology for Europe. *International Journal of Climatology*, 22.
- McKee, T. B., Doesken, N. J., Kleist, J. (1993) The relationship of drought frequency and duration to time scales. *8th Conference on Applied Climatology*, 179–184.
- Palmer, W. C. (1965) Meteorological Drought. *Office of climatology*.
- Peters, E. (2003) *Propagation of drought through groundwater systems: illustrated in the Pang (UK) and Upper-Guadiana (ES) catchments*. Wageningen Universiteit.
- Puš, V. (2011) *Popisná statistika*. Skripta ČZU.

- Tallaksen, L. M., van Lanen, H. A. J. (2004) *Hydrological Drought. Processes and Estimation Methods for Streamflow and Groundwater*. Elsevier, Amsterdam.
- Vokoun, M. (2014) *Verifikace předpovědi srážek pro hydrologické modelování*. Master's thesis, Česká zemědělská univerzita, Česká republika.
- Von Storch, H., Zwiers, F. W. (2001) *Statistical analysis in climate research*. Cambridge University Press.
- Wanders, N., Van Lanen, H., van Loon, A. F. (2010) Indicators for drought characterization on a global scale.
- Yevjevich, V., Ingenieur, J., Yevjevich, V., et al. (1967) *An objective approach to definitions and investigations of continental hydrologic droughts*. Colorado State University Fort Collins.
- Yevjevich, V., Engineer, Y., Yevjevich, V., et al. (1972) *Probability and statistics in hydrology*. Water resources publications Fort Collins, CO.

METODY VYHODNOCOVÁNÍ VODOHOSPODÁŘSKÝCH DAT

Martin Hanel, Adam Vizina

Vydala Česká zemědělská univerzita v Praze v roce 2014.

ISBN ??

Vydání první

Počet stran: ??

Náklad: ??

Tisk:??