

# Statistické prostředí R

Martin Betinec

UNIVERZITA KARLOVA V PRAZE  
Filosofická fakulta  
Katedra sociologie

27. ledna 2008

## Vznik a zařazení

### Vývoj

- ▶ S ... jazyk ⇒ SPLUS
- ▶ R ... GNU verze  
(98% kompatibilní syntaxe)

### Konkurenti

- ▶ SAS ... velká data (banky)
- ▶ SPSS ... humanitní
- ▶ NCSS ... biologie
- ▶ Statistica

### Výhody Rka

- ▶ intuitivnost modelů
- ▶ množství knihoven
- ▶ náklady ... GNU
- ▶ help
- ▶ grafika
- ▶ komunita uživatelů  
a vývojářů

## Instalace

### WWW adresy

- ▶ stránka projektu ... <http://www.r-project.org/>
  - ▶ Manualy
  - ▶ Wiki
  - ▶ FAQ
  - ▶ Download (base + balíky) ... CRAN
- ▶ český mirror CRANu ... <http://cran.biokontakt.cz/>

# A a $\Omega$ aneb spouštění a opouštění

## Začátek

- ▶ Unix/Linux-ech . . . z terminálu R
- ▶ MS Win . . . klik
- ▶ viz úvodní výpisy

## Konec

- ▶ `q()` . . . je to funkce (viz `help(q)`) – proto závorky
- ▶ ? uložení pracovního prostředí . . . proměnné

## Příklad 1. . . Spouštění a knihovny

1. Spustíte Rko. Co Vám to píše?
2. Vyzkoušejte:
  - 2.1 `demo(image)`
  - 2.2 `data()`
  - 2.3 `library()`
  - 2.4 `library(Rcmdr)`
  - 2.5 `help.start()`
3. Ukončete Rko.

## Natažení vestavěných dat a knihoven

### Data

- ▶ `data()` . . . seznam použitelných
- ▶ `data(women)` . . . load konkrétních

### Knihovny

- ▶ `library()` . . . seznam použitelných
- ▶ `library(ISwR)` . . . load konkrétních
- ▶ užitečné:  
MASS, ISwR, car, e1071, lattice, Rcmdr,  
rggobi, rattle, xtable, ROCR, kernlab,  
hints, foreign, cluster

- ▶ `library(Rcmdr)`
- ▶ klikací GUI
- ▶ ulehčení začátků
  - ▶ inspirace ... co všechno lze
  - ▶ nápověda syntaxe

## Systém nápovědy

### Možnosti nápovědy

- ▶ `help("prikaz")` ... textová nápověda
- ▶ `?prikaz` ... zkráceně totéž  
(nefunguje na klíčová slova (`for`, `while`))
- ▶ `help.start("prikaz")` ... HTML help
- ▶ `help.search("cosi")` ... vyhledá výskyt v manuálech

## Systém nápovědy II

### Struktura help stránek

- ▶ Usage
- ▶ Example ... lze spustit, rychlá orientace
- ▶ Argumenty ... input
- ▶ Value ... output
- ▶ Detaily ... podrobnosti o procedurách, reference
- ▶ See also ... pro neznámá klíčová slova  
(znám-li klíčové slovo něčeho obdobného)

### Prozkoumejte příkaz `...chisq.test`

1. K čemu se používá?
2. V jaké knihovně jej najdeme?
3. Používá klasický vzorec pro výpočet testové statistiky?

## Doporučení pro práci

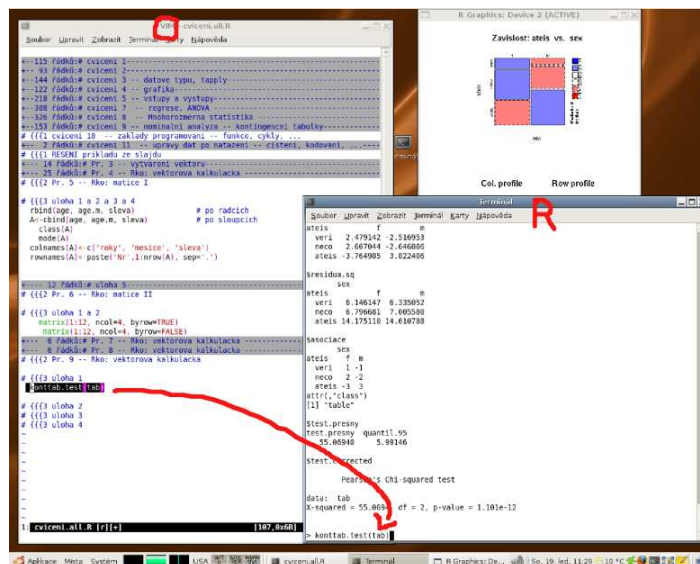
- ▶ v konzoli využívat šipek . . . historie
- ▶ kód psát zvlášť ⇒ kopírovat do konzole



### Snazší

- ▶ kopírování víceřádkových příkazů
- ▶ opravy . . . editor pohodlnější než konzole
- ▶ uložení hotového kódu
- ▶ výroba skriptů

## Obrázek pracovního prostředí



# Jak lze v Rku řešit problémy

- ▶ **programátorsky**
  - ▶  $\oplus$  ... nejobecnější (úroveň C)
  - ▶  $\ominus$  ... složité
- ▶ **matematický jazyk**
  - ▶  $\oplus$  ...  $\exists$  málo pravidel k zapamatování
  - ▶  $\ominus$  ... někdy dřevorubecké
- ▶ **statistický jazyk**
  - ▶  $\oplus$  ... intuitivní
  - ▶  $\ominus$  ... nutná znalost zaklínadel

## Příklad 3. . . vícera způsobů řešení

### Kategorizace numerického vektoru `vjek`

- ▶ programátorsky

```
vjek.2<-vjek
for(i in 1:length(vjek)) {
  if (vjek[i]<= 18) {
    vjek.2[i]<-'dite'
  } else {
    vjek.2[i]<-'dospely'
  }
}
```

## Příklad 3. . . vícera způsobů řešení

### Kategorizace numerického vektoru `vjek`

- ▶ R-kovsky

```
vjek.2<-rep('dospely', length(vjek))
vjek.2[vjek<= 18]<-'dite'
```

# Přiřazení proměnných

## Operátor přiřazení

- ▶ `<-` ... přiřazení doleva
- ▶ `->` ... přiřazení doprava
- ▶ `_` ... zastaralé, nepřehledné
- ▶ příklad:  

```
x<-5 ; vekt.x<-c(2,3,5,6)
jmena<-c('Adam', 'Bruno', 'Cyril', 'Don')
```

## Poznámka o názvech proměnných

### Jména nesmí

- ▶ začínat číslem
- ▶ obsahovat klíčová slova ani řídící znaky  
(`for`, `while`, `repeat`, `c`, `_`, `:`)
- ▶ obsahovat `☐` ... zpětná kompatibilita s SPLUS

### Doporučení

- ▶ case sensitive ... nezneužívat
- ▶ srozumitelnost ... samovysvětlující názvy, komentáře

## Základní datové třídy

### 2 axiomy

- ▶ vše v jazyku R je objekt
- ▶ každý objekt má třídu

### Základní třídy

- ▶ `character`
- ▶ `logical` ... `TRUE/FALSE`
- ▶ `numeric` ...  $\mathbb{R}$
- ▶ `integer` ...  $\pm\mathbb{N}$
- ▶ `complex`

### Speciality R

- ▶ `vector`
- ▶ `matrix` / `array`
- ▶ `factor/ordered`
- ▶ `list` ... mix tříd
- ▶ `data.frame`

## Příklad 4. . . třídy objektů

### Typy a třídy

1. Jak se jmenují základní typy veličin v SPSS?
2. Jakého typu je vektor `c(1, 2, 3, 4)`
3. Jakého typu je vektor `c('1', '2', '3', '4')`
4. Jakého typu je vektor `c(1, 'NA', 3, 4)`
5. Jakého typu je vektor `c(1, NA, 3, 4)`
6. Jakého typu je vektor `c(T, F, T, T)`

## Obecné operace s třídami

- ▶ `class/unclass` ... zjistí třídu/odstraní vyšší
- ▶ `mode` ... jak to vnitřně ukládá
- ▶ `is.trida` ... ověří typ třídy ... (`is.matrix(A)`)
- ▶ `as.trida` ... změní současnou na 'trida'  
... (`as.vector(A)`)

### Další typy tříd

- ▶ `htest` ... výsledek testů ... `C {listu,}`  
viz např. `?t.test`
- ▶ `function` ... příkazy
- ▶ `table`

## Třída `vector`

### Vytváření

- ▶ `c` ... operátor konkaténace (`x<-c(1,2,5)`)
- ▶ `rep` ... replikace (`sex<-c(rep('m',5), rep('z',7))`)
- ▶ `seq` ... posloupnost (`IQ.sit<-seq(0,150,by=.1)`)  
speciálně `x<-1:10`

### Speciální typy

- ▶ `NA` ... not available
- ▶ `letters`, `LETTERS`

### operace

- ▶ `+`, `-`, `*`, `:`, `^` ... po složkách
- ▶ `length`
- ▶ `[]` ... složky

## Příklad 5. . . vytváření vektorů

### Na zelené louce

Vytvořte vektor obsahující

1. složky 10, 9, 8, 7, 1, 2, 3, 4, 5, 6.
2. sudá čísla od -100 do +50
3. Jak zjistíte počet složek vektoru?
4.  $sex = (\underbrace{"m", "m", \dots, "m"}_{20 \times}, \underbrace{"f", "f", \dots, "f"}_{23 \times})$
5. Jak rychle zjistíte četnosti jednotlivých kategorií?

## Příklad 6. . . Vektorová kalkulačka

### Vytvořte vektor obsahující

1. mocniny  $\pi$  od 1 do 5
2. odmocniny  $e$  od 2 do 5
3. věk: 23, 20, 21, 22, 24, 22, 21, 20, 33, 25, 26, 22  
a vyjádřete
  - 3.1 věk v měsících
  - 3.2 každou druhou složku v měsících
  - 3.3 ke každému staršímu než 25 přičtěte 5
  - 3.4 vytvořte další vektor `sleva`, kde
    - ▶ mladší 26 budou označení 'ISIC'
    - ▶ ostatní ... 'NE'
4. zkontrolujte vše najednou

## Třída `matrix`, resp. `array`

### Vytváření

- ▶ `cbind`, resp. `rbind` ... spojení vektorů po sloup(řád)cích
- ▶ `matrix`, resp. `array` ... z 1 vektoru

### operace

- ▶ `+`, `-`, `*`, ... po prvcích
- ▶ `%*%`, ... maticově
- ▶ `solve` ... inverze
- ▶ `[]` ... složky

### Rkovské operace

- ▶ `dim`, `nrow`, `ncol` ... dimenze
- ▶ `dimnames`, `colnames`,  
`rownames`
- ▶ `apply` ... aplikace funkcí  
(po řád/sloup/cích)



## Příklad 7. . . matice

### Matice z vícera vektorů stejné délky

1. Spojte vektory věku, věku v měsících a slevy po řádcích.
2. Totéž po sloupcích do matice  $A$ .
3. Jakého typu je matice  $A$
4. Pojmenujte vhodně dimenze  $A$ .
5. Vyberte z  $A$ 
  - ▶ prvek z 2. řádku a 3. sloupce
  - ▶ celý 4. řádek
  - ▶ 1. až 5. složku 4. řádku
  - ▶ celý prostřední sloupec
  - ▶ všechny řádky kromě 2. a 5.

## Příklad 8. . . matice II

### Matice z jednoho vektoru (z řady čísel)

1. Vytvořte z vektoru  $1:12$  matici  $B$  o 3 řádcích a 4 sloupcích.
2. Kolik je možností, jak ji vytvořit?

### Aplikace funkcí na matici

1. Co udělá `sum(B)`?
2. Zjistěte:
  - ▶ sloupcové součty
  - ▶ řádkové průměry

## Třída `factor`, resp. `ordered`

### Vytváření

- ▶ `factor`, resp. `ordered`
- ▶ argument `levels` ... správné řazení kategorií,  
srov. Př. 9 bod 7

### operace

### Užití

- ▶ nominální
- ▶ ordinální
- ▶ `relevel`
- ▶ `sort`, `order`  
... `x[order(x)] = sort(x)`
- ▶ `rank`
- ▶ `tapply`

## Příklad 9. . . Faktory

### 1. Vytvořte vektory

```
plat=(20, 16, 50, 19, 20, 17, 25, 40),  
sex= ("m", "m", "m", "z", "z", "z", "z", "z" )
```

### 2. Je `sex` nominální? Sestrojte tabulku třídění 1.stupně.

### 3. Zjistěte průměrné stáří v jednotlivých pohlavích.

### 4. Seřadíte `plat`. A sestupně.

### 5. Zjistěte pořadové statistiky platu

### 6. Přerovnejte vektor `sex` dle pořadí platu.

### 7. Porovnejte

```
kv<-ordered(c('Mi','Lo','Hi','Hi','Lo'))  
kv.2<-ordered(c('Mi','Lo','Hi','Hi','Lo'),  
              levels=c('Lo','Mi','Hi'))
```

## Třída `list`

### Užití a vlastnosti

- ▶ mix různých objektů dohromady
- ▶ výpisy

### Vytváření

- ▶ `list`
- ▶ `split` ... rozdělí dle faktoru (kratší zápis)
- ▶ `by` ... `split` pro `data.frame`y

### operace

- ▶ `sapply`, `lapply`  
... aplikace funkce na jednotlivé složky
- ▶ `names` ... jména složek

## Příklad 10... seznamy (`list`)

### Výroba seznamu

Rozdělte `plat` dle pohlaví

1. pomocí `list` do proměnné `l.1`
2. pomocí `split`

### Aplikace funkcí na `list`

Spočtěte v listu `l.1` průměrný plat dle pohlaví, aby výsledek byl

1. `list`
2. vektor

## Příklad 10/B ... seznamy II

### Výpis složek listu

Vypište z l.1

1. názvy složek
2. druhou složku
3. složku zena
4. 3. prvek první složky
5. 1. a 3. prvek složky muz

## Příklad 10/C ... seznamy III

### Příklad listu. ... výpisy výsledků testů

1. Vytvořte kontingenční tabulku:

	pohlaví	obět	
		ANO	NE
1.	zena	200	739
	muz	213	698
2. Spočítejte test nezávislosti. Výsledek uložte do prom. t.1.
3. Jaké složky má t.1?
4. Vypište jen *p-hodnotu* a očekávané četnosti za  $H_0$ .

## Třída data.frame

### Užití a vlastnosti

- ▶ datová matice ... á la matrix
- ▶ různé třídy vektorů (stejně dlouhých) ... á la list

operace ... viz matrix a list

### Vytváření

- ▶ data.frame

- ▶ summary
- ▶ attach, detach
- ▶ tapply, by
- ▶ (-/s/l)apply ... if stejný typ znaků
- ▶ subset, split, transform ... šetří zápis

## Příklad 11...data frame

### Výroba a základní operace

1. vyrobte nominální znak `pozice= (tech, asis, sef, tech, tech, asis, ved, ved)`
2. vytvořte `data.frame` **firma** ze `plat, pozice, sex`
3. vypište názvy znaků
4. sumarizujte info o znacích
5. vypište znak `pozice` a pohlaví všemi možnými způsoby
6. oba sumarizujte dohromady
7. jak odstranit otravné volání `firma$`?

## Příklad 11/B ...data frame

### Aplikace funkcí

1. rozdělte platy ve firmě dle `pozice` všemi možnými způsoby
2. spočtěte průměrný plat dle `pozice`
3. spočtěte průměrný plat dle `pozice` a `pohlaví`
4. ? platí `tapply(...) ≡ sapply(split(...))`

### Zkracující příkazy...subset, split, transform

Zkusť bez a s použitím zkracujících příkazů

1. změňte `firma` na `firma.sc`, kde `plat` je standardizován
2. změňte `firma` na `firma.us`, kde `plat` je v US\$
3. totéž s angl. labely `sexu`
4. vyberte `pozici` a `plat`, kde `plat`  $\geq 20$

## Příklad 12... generické fce

### summary a plot

zkuste, co dělají fce `summary` a `plot` na různé typy proměnných

1. `plat` ... numerická
2. `sex` ... kategoriální
3. `l.l.` ... seznam
4. `tab` ... matice
5. `tab.tab<-as.table(tab)` ... tabulka
6. `firma` ... data frame

## Příklad 13. . . Přidávání a ubírání složek

### Přidávání

Přidejte do následujících proměnných novou složku a ulozte to jako `name_of_var.2`

1. `plat ... složku 55`
2. `sex ... složku 'm'`
3. `l.1... složku dite<-c(1:5)`
4. `tab... sloupec vrah=(10, 25)`
5. `tab.tab... totéž`
6. `firma... znak hodnoceni=1:nrow(firma)`

### Ubírání

Z předchozích typů uberte vždy poslední složku

## Načtení dat dostupných defaultně v Rku

### Data

- ▶ `data()` ... seznam dostupných datových souborů v Rku
- ▶ `data(package='MASS')` ... seznam dostupných datových souborů v balíku "MASS"
- ▶ `data(women)` ... load konkrétních

## Příklad 14. . . natažení dat dostupných v Rku

### `data swiss`

1. načtete defaultní datový soubor 'swiss' a zjistěte jeho dimenzi
2. zjistět názvy jeho proměnných a jejich význam
3. charakterizujte data; míry polohy a rozptylu
4. nakreslete vhodný graf
5. co lze říct o náboženském vyznání daných kantonů?

## Načtení externích dat ...`read.table`, `scan`

### `read.table`

- ▶ z textového souboru (`*.txt`, `*.csv`, ...)
- ▶ nutno nastavit
  - ▶ zda 1. řádek obsahuje jména sloupců ...`header=T`  
(automaticky=T if 1. řádek má o 1 sloupec méně)
  - ▶ oddělovač sloupců ...`sep = ""` (default),  
`\t` ... tabulator, `,` nebo `;` ... (soubory `*.csv`)
  - ▶ desetinný oddělovač ...`dec = "."`
- ▶ dle formátu má několik zkratk s nastavnými defaulty
  - ▶ `read.csv`, `read.delim`, ...
  - ▶ `read.spss...` (balík `foreign`)
    - ▶ `to.data.frame=TRUE`, `trim.factor.names=TRUE`
    - ▶ Linux: nefunguje diakritika – nutno překódovat

## Načtení datového souboru z SPSS (`*.sav`)

- ▶ přímo `read.spss...` (balík `foreign`)
- ▶ přes textový soubor
  - ▶ v SPSS uložit jako `Tab delimited...*.dat`
  - ▶ zachová labely pouze u stringů
  - ▶ funguje vždy

### Přímo ... na Windows

#### volby

- ▶ `to.data.frame=TRUE`, `trim.factor.names=TRUE`
- ▶ pro vynechání zbytečných mezer

## Načtení datového souboru z SPSS (`*.sav`) ... Linux

### Přímo ... na Linuxu

- ▶ volby
  - ▶ `to.data.frame=TRUE`,
  - ▶ `trim.factor.names=TRUE` nefunguje (kvůli encoding)
- ▶ zachová labely
- ▶ špatná diakritika
  - ▶ načíst (`read.spss`)
  - ▶ uložit jako `*.win.txt` (`write.table`)
  - ▶ s volbou `eol="\r\n"` ... Windowsí konec řádku
  - ▶ `recode cp1250..u8 < file.win.txt`  
`> file.utf8.txt`

## Příklad 15. . . Načtení očištěných dat (\*.txt)

### data Eurosec

1. Prohlédněte si soubor `eurosec.txt`
  - 1.1 ? oddělovač sloupců ?
  - 1.2 ? oddělovač desetinných míst ?
  - 1.3 ? oddělovač textu ?
2. Načtěte data
3. Zkontrolujte, zda se dobře načetly typy proměnných
4. Proveďte analýzu jako ve cvičení 14
5. Sumarizujte znaky `ind` a `agr` podle `system-u`

## Příklad 16. . . Načtení dat z tabul. kalkulátoru (\*.xls)

### Excell, OOcalc. . . data Aktér

1. otevřete soubor `akter.reduk.(xls/ods)`
2. uložte jako
  - ▶ **formátovaný text** ... \*.txt
  - ▶ **export textu nastavitelný** ... \*.csv
3. **pozor** na
  - ▶ skryté paznaky ... preventivní ořez tabulky
  - ▶ desítný oddělovač
4. zkontrolujte a načtěte textový soubor
5. analyzujte
  - 5.1 správné načtení typů a jmen proměnných
  - 5.2 spec.: *jak se načetly ordinální veličiny?*

## Příklad 17. . . Načtení dat z SPSS (\*.sav)

### Obecně. . . přes textový soubor

1. v SPSS uložte jako **Tab – delimited** ... \*.dat
2. načtěte `studenti.SPSS.dat`
3. co se stalo s kategoriálními proměnnými?

### Přímo z SPSS... read.spss

1. Načtěte data `studenti.SPSS.sav`
2. Zkuste všechny možnosti argumentů
3. Zkontrolujte typy a proveďte třídění 1. stupně
4. Co by to chtělo?

# Načtení skriptu

## source

- ▶ pohodlnější a rychlejší spouštění (oproti klikání)
- ▶ umožňuje neinteraktivní spouštění
- ▶ `source( "~/Rko/skripty/muj.script.R" )`
- ▶ v cestě se používají `/` i ve Windows

## Příklad 18. . . výroba načítacího skriptu

### data akter

vytvořte skript `nacti.aktera.R`,

1. který načte soubor `akter.reduk.csv`
2. u ordinálních znaků opraví jejich typ
3. co by to chtělo?

## Výstupy na obrazovku

### Interaktivně

- ▶ v konzoli . . . prostě napíšu
- ▶ `> x`  
`[1] 1.17 0.58 -2.12`
- ▶ `> (x <- rnorm(3))`  
`[1] 1.17 0.58 -2.12`

### Neinteraktivně

- ▶ funkce, cykly, skripty
- ▶ `print`
  - ▶ na samostatný řádek
  - ▶ nelze kombinovat s textem
- ▶ `cat`
  - ▶ do 1 řádku
  - ▶ lze kombinovat s textem



## Příklad 19. . . výstupy na obrazovku

### Srovnání ne~/závorkovaného vstupu

Zkuste:

- ▶ `pom<-rnorm(3)`
- ▶ `(pom<-rnorm(3))`

### Srovnání `cat` a `print`

Zkuste:

- ▶ `mean(esc$tran)`
- ▶ `cat(mean(esc$tran))`
- ▶ `print(mean(esc$tran))`
- ▶ `cat("Prumer: ", mean(esc$tran), "\n")`

## Příklad 20. . . výroba načítacího skriptu II

### Zlepšení skriptu `nacti.aktera.R`

1. načte soubor `akter.reduk.csv`
2. u ordinálních znaku opraví jejich typ
3. vypíše
  - ▶ cestu k souboru
  - ▶ jméno proměnné, do níž jsme data načetli (`akt`)
  - ▶ dimenzi (`akt`)
  - ▶ názvy řádků
  - ▶ názvy proměnných podle jejich typů

## Uložení výstupu

### Plochý text . . . `sink`

- ▶ přesměruje výstup z konzole  
(popř. rozdvojí, if `split = TRUE`)
- ▶ umožňuje neinteraktivní práci
- ▶ `sink("~/Rko/vysledky/vysledek.txt")`
- ▶ nutno na závěr přesměrovat zpět . . . `sink()`
- ▶ přidání dalšího textu do téhož souboru . . . `append=TRUE`

## Příklad 21. . . uložení výsledků výpočtů

### data akter

1. charakterizujte soubor `akt` pomocí měř polohy
2. výsledky přesměrujte do souboru `vysl.akter.txt`
3. zkontrolujte
4. přidejte do téhož souboru popis variability znaků
5. zkontrolujte

### Zlepšení skriptu `nacti.aktera.R`

Ať skript to, co na obrazovku, vypisuje i do souboru `vysl.akter.txt`.

## Uložení výstupu II – fromátovaný výstup

### Tabulka v $\text{\LaTeX}$ u , resp. v HTML ...`xtable`

- ▶ `library(xtable)`
- ▶ lze rovnou do souboru
- ▶ umožňuje vložit  $\text{\LaTeX}$ ovské popisky
  - ▶ `caption, label, align`
  - ▶ po konverzi do HTML jsou také konvertovány (`anchor,...`)
- ▶ `print(xtable(tabulka), type='html') ... HTML`
- ▶ parametr `digits`
  - ▶ počet desetinných míst
  - ▶ délka = počet sloupců + 1

## Uložení výstupu III – datová matice

### `write.table`

- ▶ inverze k `read.table` ... podobné volby
- ▶ navíc `eol = "\n"` ... konec řádku
- ▶ navíc `eol = "\r\n"` ... konec řádku ve Win
- ▶ opět existují zkratky ... viz `help`
  - ▶ `write.csv`
  - ▶ `write.csv2` ... pro CZ spreadsheetsy

## Příklad 21... uložení dat

### Do textového formátu .....data eurosec

1. vyrobte novou proměnnou indikující silně průmyslové státy (i.e. `ind > 30`)
2. přidejte ji k datovému souboru `eurosec`
3. výsledek uložte jako `csv`, tj.
  - ▶ oddělení ;
  - ▶ desetinná čárka
  - ▶ sloupec řádkových názvů má jméno (prázdný znak)

### Do tagovaných formátů HTML, $\text{\LaTeX}$ .....data eurosec

výsledek uložte také jako `*.html` a jako `*.tex`

oboje zkontrolujte

## Příklad 22... Kontrola uložení `esc.new` do $\text{\LaTeX}$ -u

	agr	min	ind	engr	const	Sl	fin	serv	tran	komun	system	prumysl
BEL	3.30	0.90	27.60	0.90	8.20	19.10	6.20	26.60	7.20	nekom	demo	Low
DK	9.20	0.10	21.80	0.60	8.30	14.60	6.50	32.20	7.10	nekom	demo	Low
F	10.80	0.80	27.50	0.90	8.90	16.80	6.00	22.60	5.70	nekom	demo	Low
DDR	6.70	1.30	35.80	0.90	7.30	14.40	5.00	22.30	6.10	kom	kom	High
IRL	23.20	1.00	20.70	1.30	7.50	16.80	2.80	20.80	6.10	nekom	demo	Low
IT	15.90	0.60	27.60	0.50	10.00	18.10	1.60	20.10	5.70	nekom	demo	Low
LUX	7.70	3.10	30.80	0.80	9.20	18.50	4.60	19.20	6.20	nekom	demo	High
NL	6.30	0.10	22.50	1.00	9.90	18.00	6.80	28.50	6.80	nekom	demo	Low
UK	2.70	1.40	30.20	1.40	6.90	16.90	5.70	28.30	6.40	nekom	demo	High
AUS	12.70	1.10	30.20	1.40	9.00	16.80	4.90	16.80	7.00	nekom	demo	High
FIN	13.00	0.40	25.90	1.30	7.40	14.70	5.50	24.30	7.60	nekom	demo	Low
GRE	41.40	0.60	17.60	0.60	8.10	11.50	2.40	11.00	6.70	nekom	dikt	Low
NOR	9.00	0.50	22.40	0.80	8.60	16.90	4.70	27.60	9.40	nekom	demo	Low
POR	27.80	0.30	24.50	0.60	8.40	13.30	2.70	16.70	5.70	nekom	dikt	Low
ESP	22.90	0.80	28.50	0.70	11.50	9.70	8.50	11.80	5.50	nekom	dikt	Low
SWE	6.10	0.40	25.90	0.80	7.20	14.40	6.00	32.40	6.80	nekom	demo	Low
SWIS	7.70	0.20	37.80	0.80	9.50	17.50	5.30	15.40	5.70	nekom	demo	High
TUR	66.80	0.70	7.90	0.10	2.80	5.20	1.10	11.90	3.20	nekom	dikt	Low
BUL	23.60	1.90	32.30	0.60	7.90	8.00	0.70	18.20	6.70	kom	kom	High
CZ	16.50	2.90	35.50	1.20	8.70	9.20	0.90	17.90	7.00	kom	kom	High
D	4.20	2.90	41.20	1.30	7.60	11.20	1.20	22.10	8.40	nekom	demo	High
HUN	21.70	3.10	29.60	1.90	8.20	9.40	0.90	17.20	8.00	kom	kom	Low
PL	31.10	2.50	25.70	0.90	8.40	7.50	0.90	16.10	6.90	kom	kom	Low
ROM	34.70	2.10	30.10	0.60	8.70	5.90	1.30	11.70	5.00	kom	kom	High
CCCP	23.70	1.40	25.80	0.60	9.20	6.10	0.50	23.60	9.30	kom	kom	Low
YUG	48.70	1.50	16.80	1.10	4.90	6.40	11.30	5.30	4.00	kom	dikt	Low

## Uložení výstupu IV – grafický výstup

- ▶ přesměrování grafického výstupu do souboru
- ▶ dle formátu:
  - ▶ `postscript ... soubory *.eps`
  - ▶ `pdf... *.pdf`
  - ▶ `jpeg... *.jpeg`
  - ▶ `png... *.png`
- ▶ volby
  - ▶ `file ... kam to zapíše`
  - ▶ `width, height...`
    - ▶ v palcích (`postscript, pdf`)
    - ▶ v pixelech (`jpeg, png`)
  - ▶ `horizontal=FALSE ... postscript`
  - ▶ `onefile = FALSE ... pro animace`
- ▶ `dev.off()` ... zpětné přesměrování na obrazovku

## Příklad 23. . . uložení grafiky

```
boxplot .....data eurosec
```

1. nakreslete boxploty kardinálních znaků
2. uložte obrázek do formátů
  - ▶ postscript
  - ▶ PDF
  - ▶ jpeg
  - ▶ png
3. výsledky zkontrolujte

## Grafika v R-ku – Témata

- ▶ příkazy + jejich parametry
  - ▶ 1. úrovně ... vždy nový graf (`plot`)
  - ▶ 2. úrovně ... přidává do aktuálního grafu (`points`)
- ▶ speciální grafy (`boxplot`)
- ▶ rozvržení obrázku ... (1 obr. v více) v 1 grafu
- ▶ popisky a legendy
- ▶ dynamické obrázky

## Příkaz `plot`

### Kreslí

- ▶ objekt ... *generic function* (`plot.lm`)
- ▶ souřadnice 2D ... `plot(x,y,...)`
- ▶ model ... `plot(y ~ x,...)`

### Další argumenty – viz `?par`

- |   |   |
|---|---|
| ▶ typ čáry <code>type</code> <ul style="list-style-type: none"><li>▶ 'p' ... points (default)</li><li>▶ 'l' ... lines</li><li>▶ 'b' ... both</li><li>▶ 'n' ... none</li></ul> | ▶ meze ... <code>xlim, ylim</code> <ul style="list-style-type: none"><li>▶ popisky<ul style="list-style-type: none"><li>▶ <code>main</code> ... hlavní</li><li>▶ <code>sub</code> ... podnadpis</li><li>▶ <code>xlab, ylab</code> ... osy</li></ul></li></ul> |
|---|---|

## Příkazy nižší úrovně

přidávají do grafu

- ▶ `lines` ... lineár. interpolace
- ▶ `points`
- ▶ `abline` ... přímka
- ▶ `rect` ... obdélník
- ▶ `polygon`, `segment` ... složitější oblasti
- ▶ `text` ... popisky
- ▶ `arrows` ... šipky

## Argumenty fce `plot`

### Vlastnosti čáry

- ▶ `lty` ... typ
  1. plná (default)
  2. čárkovaná
  3. tečkovaná
  4. čerchovaná
  5. ...
- ▶ `lwd` ... šířka

### Vlastnosti bodů

- ▶ `pch`
  1. kroužky
  2. trojúhelníčky
  3. křížky
  4. křížky šikmé
  5. ...

## Barvy

- ▶ lze zadat číslem
  1. černá (default)
  2. červená
  3. zelená
  4. modrá
- ▶ lze i slovně ("`darkorange`", "`navy`")
- ▶ viz `palette`

# Interaktivní prvky

`locator`

- ▶ zaměření myší
- ▶ lze i přiřadit

`identify`

- ▶ myší se klikne na pozorování, která chceme identifikovat

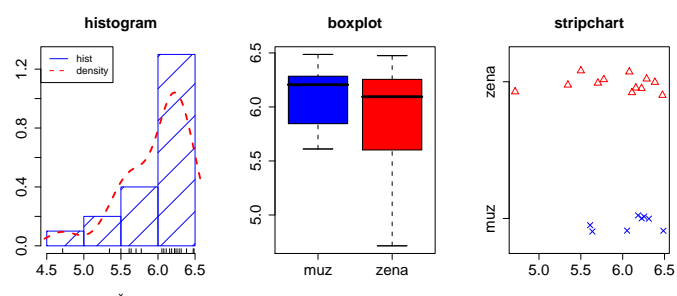
## Více obrázků v jednom grafu

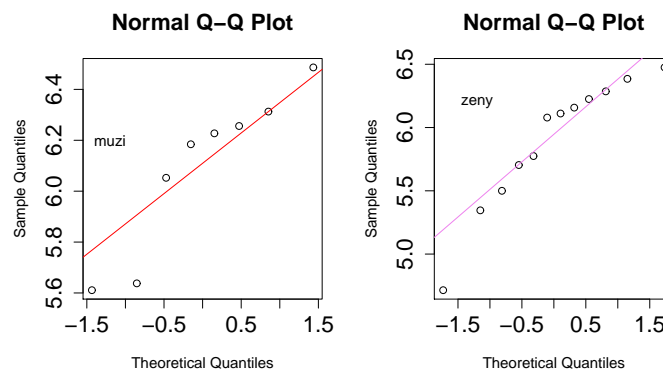
argumenty nastavení vlastností obrázku `... par`

- ▶ stejně velké podobrázky
  - ▶ `mfc col`
  - ▶ `mf row`
- ▶ různě velké
  - ▶ `layout ...` samostatný příkaz
  - ▶ `fig ...` parametr `par`

## Speciální grafy – kardinální

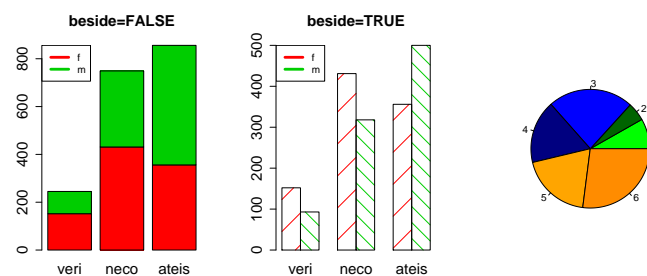
- ▶ `hist ...` histogram
- ▶ `boxplot`
- ▶ `stripchart ...` body ve skupinách
- ▶ `qqnorm`, `qqline`, `qqplot ...` Q-Q diagramy



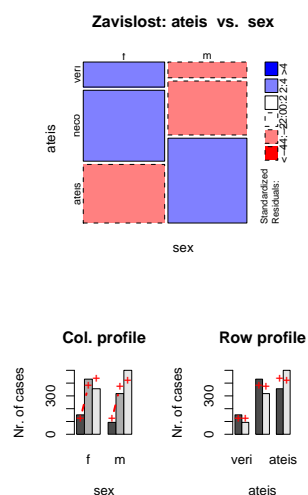


## Speciální grafy – kategoriální

- `barplot` ... tyčový diagram
- `pie` ... koláč
- `mosaicplot` ... vícerozměrný



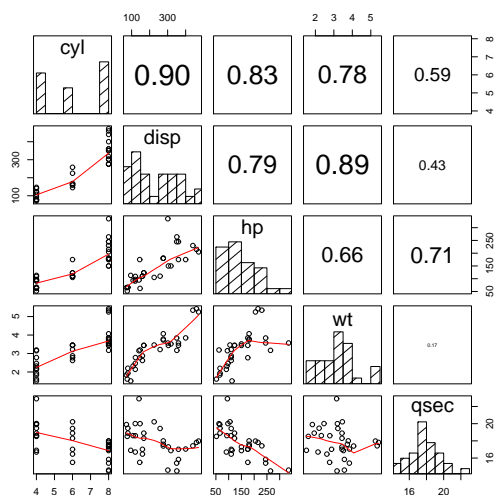
## Speciální grafy – kategoriální – mozaikový diagram



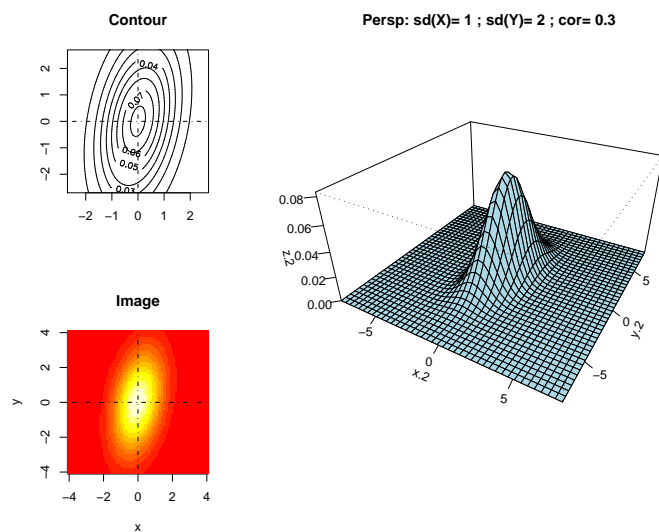
# Speciální grafy – mnohorozměrné

- ▶ `pairs` ... matice *scatter plotů*
- ▶ `contour` ... vrstevnicový graf
- ▶ `image` ... 3. dimenze barvami
- ▶ `persp` ... 3D zobrazení

## Mnohorozměrné grafy – matice scatterplotů (`pairs`)



## Mnohorozměrné grafy





# Úlohy

1. Nakreslete do jednoho obrázku grafy funkcí:

1.1  $f: y = \sin(x)$

1.2  $g: y = \sin(2x)$

1.3  $h: y = \sin(3x)$

1.4  $j: y = \sin(x/2)$

v rozsahu od  $-2\pi$  do  $+2\pi$  a dodejte legendu.

2. Nakreslete graf složený ze dvou obrázků:

2.1 vlevo: graf z bodu 1

2.2 vpravo graf funkce

$$m: y = 0.1 \sin\left(\frac{x}{2}\right) + 0.2 \sin(x) + 0.3 \sin(2x) + 0.4 \sin(3x)$$

## Psaní vlastních funkcí

- ▶ matematické ... kreslení
- ▶ zlepšení nabízených ... `my.pairs`
- ▶ šetření kódu + konceptualizace ... R-poetry
- ▶ aplikace v `(-/t/s/l)apply` ... úpravy dat

## Struktura funkcí

```
jméno.funkce <- function(arg.1, arg.2,...){  
  tělo funkce  
    
    
    
    
}
```

## IN ... Argumenty

- ▶ specifikované defaultně
- ▶ nespecifikované
- ▶ neuvedené ...
- ▶ `args...` výpis argumentů funkce

## OUT

`return ... list`

## Příklad 24. ... Psaní vlastních funkcí

### Napište funkce pro

1. výpočet kořenů kvadratické rovnice
2. spočtení směrodatné odchylky
3. tabulku třídění 1. stupně, kde budou
  - ▶ absolutní četnosti
  - ▶ relativní četnosti
  - ▶ součty

## Kontrola proměnných

### Čištění

- ▶ `missingy ... is.na, is.element`
- ▶ `typy ... class`
- ▶ `stavy ... table`
- ▶ užití vlastních funkcí

## Příklad 25. . . Kontrola proměnných

### missingy

Zkontrolujte u data framu, zda tam jsou řádky s missingy

1. `z[is.element(z,c(9,99,999))]<-NA`
2. `kttere.row<- which(apply(is.na(dat),1,sum)>0)`

### typ a stavy proměnných . . . akter

1. Zkontrolujte typ všech proměnných
2. Zkontrolujte u všech proměnných jejich kategorie
3. ? Je v pořádku proměnná `obyvatel`
4. Nakreslete obrázky každé z proměnných

## Úprava proměnných

- ▶ `transform` . . . na data framy
- ▶ `cut` . . . kategorizace
- ▶ `levels` . . . stavy kategoriálních proměnných
- ▶ `recode` . . . balík `car`
- ▶ `symnum` . . . symbolicky kóduje číselné rozsahy

## Příklad 26. . . Úprava proměnných

### Kategorizace . . . akter

1. Vhodně kategorizujte věk
2. Jak to provede automaticky, při zadání počtu kategorií
3. zkuste `symnum(cor(akt[,13:20]))`

# Základní testy I

## Testy o střední hodnotě (o poloze)

- ▶ `t.test` ... t-test (párový i nepárový)
- ▶ `wilcox.test` ... Wilcoxon-Mann-Whitney
- ▶ `aov` ... ANOVA
- ▶ `kruskal.test` ... neparametrická ANOVA
- ▶ `friedman.test` ... neparametrická ANOVA  
s náhodnými bloky

# Základní testy II

## Testy o souvislosti znaků

- ▶ `cor.test` ... test korelačního koeficientu  
(Pearson, Kendall, Spearman)
- ▶ `chisq.test` ... Pearsonův  $\chi^2$  pro 2D kontingenční tab.
- ▶ `mcnemar.test` ... test symetrie

# Základní testy III

## Testy shody rozptylu

- ▶ `var.test` ... Fisherův
- ▶ `levene.test` ... Leveneův
- ▶ `bartlett.test` ... Bartlettův

## Ověřování rozdělení

- ▶ `shapiro.test` ... Shapiro-Wilk Normality Test
- ▶ `ks.test` ... Kolmogorov-Smirnov Test

# Modely

## Druhy modelů

- ▶ `lm` ... lineární regrese
- ▶ `glm` ... zobecněné lineární modely
  - ▶ logistická regrese
  - ▶ loglineární modely pro kontingenční tabulky
- ▶ `aov` ... ANOVA

## Zápis modelu

$y \sim x.1 + x.2 + \dots + x.n$

- ▶  $y$  ... závislá proměnná
- ▶  $x.1, x.2, \dots, x.n$  ... nezávislé
- ▶ příklady
  - ▶  $y \sim x.1 + x.2$  ... model bez interakcí
  - ▶  $y \sim x.1 + x.2 + x.3 + x.1:x.2$   
... model s interakcemi mezi  $x.1$  a  $x.2$
  - ▶  $y \sim x.1 * x.2$  totéž jako  
 $y \sim x.1 + x.2 + x.1:x.2$

## Příklady modelů

### Lineární regrese ... posloupnost hierarchických modelů

1. `mod.1<-lm(serv~agr*tran)`
2. `mod.2<-lm(serv~agr+tran)`
3. `mod.3<-lm(serv~agr)`

### Logistická regrese

```
mod.2<-glm(komun~agr + ind + fin, data=esc,  
            family=binomial("logit"))
```

## Logneární modely . . . pro 3D tabulku

- ▶ `glm(pocet ~ obet+vrah+trest, family='poisson')`  
uplné nezávislosti . . . . . [O] [V] [T]
- ▶ `glm(pocet ~ obet + trest*vrah, family='poisson')`  
sdružené nezávislosti . . . . . [O] [VT]
- ▶ `glm(pocet ~ obet*trest + obet*vrah, family='poisson')`  
podmíněné nezávislosti . . . . . [OT] [OV]
- ▶ `glm(pocet ~ vrah*trest + obet*vrah + obet*trest, family='poisson')`  
párové závislosti . . . [VT] [VO] [TO]
- ▶ `glm(pocet ~ obet*vrah*trest, family='poisson')`  
saturovaný . . . . . [VTO]

## Funkce pro práci s modely

často odpovídají složkám modelu (výsledek = list)

- ▶ `summary`
- ▶ `předpověď dle modelu`
  - ▶ `fitted` . . . v naměřených bodech
  - ▶ `predict` . . . v libovolném novém bodě
- ▶ `deviance` . . . reziduální součet čtverců
- ▶ `resid(residuals)` . . . rezidua
- ▶ `coef` . . . regresní koeficienty
- ▶ `anova` . . . porovnávání hierarchických modelů
- ▶ `plot` . . . standardní diagnostické obrázky

## Mnohorozměrná statistika

### Průzkumové analýzy

- ▶ `prcomp` . . . an. hlavních komponent
- ▶ `factanal` . . . faktorová an.
- ▶ `shluková analýza`
  - ▶ `hclust` . . . hierarchická
  - ▶ `kmeans, pam` . . . nehierarchická
- ▶ `lda` . . . lineární diskriminační an.
- ▶ `corresp` . . . korepondenční an.
- ▶ `cmdscale` . . . mnohorozměrné škálování