

Obecné poznámky, nápověda

- R rozlišuje velikost písmen.
- Symbol `.` nemá zvláštní syntaktický význam, používá se místo mezery podobně jako `_` v jiných jazycích.
- NA: not available - výraz pro chybějící hodnotu
- `help(funkce)` ; `?funkce`: vyvolání nápovědy pro funkci `funkce`

Práce s balíčky

- `install.packages(package)` : instalace balíku jménem `package`
- `library(package)` : načte balíček `package`

Základní operace

- `3+4`; `1/5`; `2^5`; `sin(pi/2)`; `log(2)`; `1/(2+3)`, ...
Běžné matematické operace včetně použití funkcí lze v R provádět zcela intuitivně.
- `a<-5`; `5->a`; `a=5`: přiřazení; všechny zápisy mají stejný význam
- `a<b`; `a==b`; `a>=b`: operátory porovnání
- `paste(...)`: spojení řetězců

Práce s vektory

- `v<-c(3.2, 2, 1.4, -1.5, 0.5)`; `w<-c("a", "b", "a", "c")`: vytvoření vektoru ze zadaných hodnot
- `seq(a, b, c)`: posloupnost od `a` do `b` s krokem `c`
- `seq(a, b, len=k)`: posloupnost od `a` do `b` z `k` členů
- `a[3]`; `a[2:5]`; `a[c(1, 3, 5)]`: výběr prvků z vektoru (3.; 2. až 5.; 1., 3. a 5. prvek); prvky jsou indexovány od 1.
- Aritmetické operace s vektory jsou prováděny po prvcích. Je-li jeden z vektorů kratší, je "periodicky prodloužen" na délku delšího vektoru.
- `length(a)`: počet prvků vektoru

Práce s daty

- `read.table(file, header = FALSE, sep = "", ...)`: načte data z textového souboru; `header = TRUE`, pokud první řádek obsahuje názvy proměnných; `sep`: oddělovací znak; je-li `sep=""`, jsou oddělovačem bílé znaky
- `read.delim(filename)`: načte data oddělená tabulátorem ze souboru
- `factor(a)`: vytvoří z vektoru faktor
- `read.delim("clipboard")`: načte data ze schránky, vhodné pro zadání dat z tabulkového procesoru
- `factor(a)`: vytvoří z vektoru faktor (vektor s hodnotami kategoriální veličiny)
- `data.frame(a, b, c, ...)`: vytvoří data frame ze zadaných vektorů (každý vektor obsahuje hodnoty jedné veličiny)
- `data1<-fix(data)`: otevře tabulkový editor data frame `data`; vrátí editovaná data a uloží do data frame `data1`
- `head(data)`; `tail(data)`: zobrazí začátek (resp. konec) data frame
- `attach(data)`: zpřístupní sloupce v data frame jako proměnné

- `stack (data)` : datové sloupce v data frame `data` spojí do jednoho sloupce a přidá faktor s názvy původních sloupců (převod ze sloupcového formátu dat do standardního datového formátu)
- `unstack (data)` : převod ze standardního datového formátu do sloupcového
- Obecně je sloupec v data frame `data` přístupný zápisem `data$sloupec`.

Informace o datech

- `names (data)` : jména proměnných v data frame
- `is.na`: testuje výskyt NA
- `nrow (data)` : počet řádků v data frame
- `ncol (data)` : počet sloupců v data frame

Základní statistické funkce

- `sum`: součet prvků vektoru
- `mean`: výběrový průměr
- `median`: medián
- `var`: výběrový rozptyl
- `sd`: výběrová směrodatná odchylka
- `min`, `max`, `range`: minimum, maximum, rozsah (vektor (min, max))

Souhrnné charakteristiky

- `summary`: souhrnné charakteristika, výstup závisí na typu argumentu
- `tapply (x, index, function)`: vrátí výsledky funkce `function` aplikované na všechny hodnoty proměnné (vektoru) `x`, které odpovídají jednotlivým kombinacím faktorů v seznamu `index`.
- `table (f1, f2)` : vrátí počty pozorování pro jednotlivé kombinace faktorů

Grafy

- `plot (a)` : vykreslí hodnoty z vektoru `a` (osa y) v závislosti na pořadí (osa x); je-li `a` faktor, kreslí sloupcový graf
- `plot (x, y)` : bodově vykreslí hodnoty `y` (osa y) proti `x` (osa x); `plot (x, y, type='l')` : obdobně vykreslí čárový graf; příkaz `plot` má celou řadu volitelných parametrů; viz `help`
- `points`: kreslení jednotlivých bodů do stávajícího grafu
- `lines`: kreslení úseček do stávajícího grafu
- `hist`: histogram
- `boxplot (v)` : krabicový graf
- `boxplot (v~a)` : boxploty proměnné `v` pro každou hodnotu faktoru `a`
- `library (lattice)` ; `xyplot (a~b|c)` vykreslí závislost `a` na `b` pro každou hodnotu faktoru `c`
- `barplot (v)` : sloupcový graf vektoru `v`
- `barplot (t)` : sloupcový graf pro tabulku `t`
- `mosaicplot (t)` : mozaikový graf pro tabulku `t`

- `pie(v)`: koláčový graf
- Pro inspiraci: `demo(graphics)`; `demo(image)`
- Postup pro uložení vytvořeného grafu

```
png('obrazek.png', width=800, height=600): vytvoří soubor pro uložení obrázku ve formátu png;
obdobně pro jpg, bmp, tiff, ...
plot(...), points(...), lines(...), ...: vykreslí graf
dev.off(): uloží výsledný graf do souboru
```

Rozdělení pravděpodobnosti

- Pro každé běžně používané rozdělení pravděpodobnosti existují v R 4 příkazy začínající na `d`, `p`, `q`, `r` pro pravděpodobnostní funkci (nebo hustotu pravděpodobnosti), distribuční funkci, kvantilovou funkci a generování vzorků.
- `dnorm(x, mean, sd)`: hustota pravděpodobnosti normálního rozdělení s parametry `mean` (stř. hodnota) a `sd` (směrodatná odchylka); `x` může být vektor
- `pnorm(x, mean, sd)`: distribuční funkce normálního rozdělení
POZOR: V R je distribuční funkce náhodné veličiny X definována jako $F_X(x) = P(X \leq x)$.
- `qnorm(x, mean, sd)`: kvantilová funkce (inverzní k distribuční funkci) normálního rozdělení
- `rnorm(n, mean, sd)`: vygeneruje `n` náhodných vzorků z normálního rozdělení
- obdobně pro další rozdělení: `d, p, q, r + binom` (binomické), `geom` (geometrické), `hyper` (hypergeometrické), `nbinom` (negativně binomické), `pois` (Poissonovo), `exp` (exponenciální), `gamma` (gamma), `weibull` (Weibullovo), ...

POZOR: Veličina s *geometrickým* rozdělením je v R definována jako *počet neúspěchů* před 1. úspěchem a veličina s *negativně binomickým rozdělením* jako *počet neúspěchů* před *k*. úspěchem.

- *Příklad* (graf hustoty): `x=seq(-5, 5, len=100)`; `y=dnorm(x, 0, 1)`; `plot(x, y, type='l')`; vykreslí graf hustoty normálního rozdělení s parametry 0 a 1; `type='l'` zajistí spojení bodů v grafu čarou
- *Příklad* (graf hustoty a pravděpodobnostní funkce): `x=seq(0, 10, len=100)`; `y=dnorm(x, 5, 2.5*0.5)`; `plot(x, y, type='l')`; `x=seq(0, 10)`; `y=dbinom(x, 10, 0.5)`; `points(x, y)`: vykreslí graf hustoty normálního rozdělení spojitou čarou a do něj graf pravděpodobnostní funkce binomického rozdělení

Intervalové odhady, testy hypotéz

- `t.test(data, mu=41)`: oboustranný t-test nulové hypotézy $H_0: \mu = 41$, vrací i 95% konfidenční interval
- `t.test(data, conf.level=0.99, alternative='l')`:
levostranný test na hl. významnosti 1%, `alternative=... 'l'` ('less') - pravostranný, 'g' ('greater') -
levostranný, 't' ('two.sided') - oboustranný
- `t.test(data1, data2, var.equal=T)`: dvouvýběrový t-test pro shodné rozptyly, `var.equal=F`: Welchův t-test pro rozdílné rozptyly
- `prop.test(x, n, alternative=..., conf.level=...)`: test a intervalový odhad parametru v binomickém rozdělení
- `wilcox.test(data, mu=41)`: jednovýběrový Wilcoxonův test mediánu symetrického rozdělení
- `wilcox.test(data1, data2)`: Mannův Whitneyův test
- `shapiro.test(x)`: Shapiro-Wilkův test normality

ANOVA

- `a=aov(x~y)`: analýza rozptylu, x : spojitá veličina, y : faktor; výsledek uloží do `a` pro další zpracování
- `summary(a)`: zobrazí výsledek ANOVy; `a` je výsledkem funkce `aov`
- `TukeyHSD(a)`: post-hoc analýza pomocí Tukeyovy HSD metody; `a` je výsledkem `aov`
- `plot(TukeyHSD(a))`: vykreslí intervalové odhady rozdílů středních hodnot `a` je výsledkem `aov`
- `oneway.test(x~y)`: ANOVA s Welchovou korekcí pro normální data s nestejnými rozptyly
- `kruskal.test(x~y)`: Kruskal-Wallisův test
- `dunn.test(x, y, method='bonferroni')`: post-hoc analýza pomocí Dunnové metody s Bonferroniho korekcí; (*balíček dunn.test*)
- `bartlett.test(x~y)`: Bartlettův test rovnosti rozptylů n.v. x pro jednotlivé hodnoty faktoru y ; předpokládá normalitu dat

Lineární regrese

- `m=lm(y~x)`: vytvoří lineární model pro vysvětlovanou veličinu y a vysvětlující veličinu x ; výsledek uloží do proměnné `m`
- `summary(m)`: vypíše informace k vytvořenému lineárnímu modelu `m` - odhady koeficientů, celkový F-test, dílčí t-testy, koeficient determinace
- `plot(x, y)`: bodový graf
- `abline(m)`: vykreslí regresní přímku lineárního modelu `m`
- `confint(m)`: konfidenční intervaly pro odhady koeficientů modelu
- `predict(m)`: predikované hodnoty pro model `m`; `predict(m, list(x=c(...)))`: predikce pro zadané hodnoty vysvětlující veličiny; `predict(m, list(x=c(...)), interval='predict')`: pás predikce v zadaných bodech; `predict(m, list(x=c(...)), interval='confidence')`: pás spolehlivosti v zadaných bodech
- `resid(m)`: rezidua
- grafické nástroje pro základní diagnostiku modelu: `par(mfrow=c(2,2))`: připraví grafický výstup pro 4 grafy; `plot(m)`: vykreslí diagnostické grafy