

Materiały do zajęć 3 z Programowania narzędzi analitycznych

1. Wykresy

Wybrane funkcje do tworzenia wykresów:

`plot` - tworzy wykresy: liniowe lub rozproszenia

`hist` - histogram

`barplot` - wykres słupkowy

`boxplot` - wykres pudełkowy

`qqnorm` - wykres kwantylowy

`curve(x^2, from=0, to=2)` - tworzy wykres krzywej określonej wzorem, argumentem jest x

2. Dodatkowe elementy wykresów

`abline(3, 5)` - dodaje do wykresu linię określoną wzorem $y = 3 + 5x$

`abline(v = 2)` - dodaje do wykresu linię pionową, $x = 2$

`abline(h = 0)` - dodaje do wykresu linię poziomą, $y = 0$

`lines` - dodaje do wykresu dodatkową linię/wykres

`points` - dodaje punkty

`text` - dodaje tekst

`title` - dodaje tytuł

3. Przykłady

```
plot(table(rpois(100, 5)), type = "h", col = "red", lwd = 10,
      main = "rpois(100, lambda = 5)")
```

```
plot(-1:1, -1:1, type = "n", xlab = "Re", ylab = "Im")
K <- 16; text(exp(1i * 2 * pi * (1:K) / K), col = 2)
```

```
plot(States03$Unemp, States03$Poverty, xlab = "Unemployment", ylab = "Poverty")
plot(Poverty ~ Unemp, data = States03, xlab = "Unemployment", ylab = "Poverty")
hist(States03$Poverty, main = "Poverty", xlab = "percent",
     xlim = c(0, 24), ylim = c(0, 20))
plot(1:19, 1:19, pch = 1:19, xlab = "x", ylab = "y")
pie(rep(1, 8), col = 1:8)
curve(x^2, from = 0, to = 2)
curve(cos(x), from = 0, to = pi)
curve(cos(x), from = 0, to = pi, lty = 4, col = "red")
```

3. Opcje wykresów

`pch` - styl znaczników

`lty` - typ linii, Line types can either be specified as an integer (0=blank, 1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses 'invisible lines' (i.e., does not draw them).

`lwd` - grubość linii

`col` - kolor, (`col='red', 'blue', ...`)

`xlim` - zakres osi x: `xlim=c(min, max)`

`ylim` - zakres osi y

`xlab` - etykieta osi x: `xlab='moja etykieta'`

`ylab` - etykieta osi y

`main` - główny tytuł

`sub` - podtytuł

Na podstawie Mathematical Statistics with Resampling and R ([link](#)).

Materiały do zajęć 2 z Programowania narzędzi analitycznych

1. Statystyka opisowa

`sum(wektor)` - suma elementów wektora lub macierzy
`mean(wektor)` - wylicza średnią z wektora
`median(wektor)` - wyznacza medianę z wektora
`sd(wektor)` - wylicza odchylenie standardowe wartości wektora
`var(wektor)` - wylicza wariancję wartości wektora
`abs(liczba)` - wylicza moduł/wartość absolutną liczby
`quantile(x, probs=0.25)` - wylicza pierwszy kwartył

2. Moda/dominanta

```

Mode <- function(x, na.rm = FALSE) {
  if(na.rm){
    x = x[!is.na(x)]
  }

  ux <- unique(x)
  return(ux[which.max(tabulate(match(x, ux)))] )
}
  
```

Źródło: link: <https://stackoverflow.com/>

3. Kwantyle

`qnorm(x, mean=0, sd=1)` - kwantyle rozkładu standardowego normalnego
`pnorm(x, mean=0, sd=1)` - wartość dystrybucyjny rozkładu w punkcie x , tj. $\Phi(x)$

Rozkład	Kwantyl	Gęstość	Dystrybuanta	Liczby losowe
Normalny	<code>qnorm(p, mean, sd)</code>	<code>dnorm(x, mean, sd)</code>	<code>pnorm(q, mean, sd)</code>	<code>rnorm(n, mean, sd)</code>
Beta	<code>qbeta(p, s1, s2)</code>	<code>dbeta(x, s1, s2)</code>	<code>pbeta(q, s1, s2)</code>	<code>rbeta(n, s1, s2)</code>
χ_n^2	<code>qchisq(p, df)</code>	<code>dchisq(x, df)</code>	<code>pchisq(q, df)</code>	<code>rchisq(n, df)</code>
Wykładniczy	<code>qexp(p, rate)</code>	<code>dexp(x, rate)</code>	<code>pexp(q, rate)</code>	<code>rexp(n, rate)</code>
t-Studenta	<code>qt(p, df)</code>	<code>dt(x, df)</code>	<code>pt(q, df)</code>	<code>rt(n, df)</code>
Jednostajny(0,1)	<code>qunif(p, min, max)</code>	<code>dunif(x, min, max)</code>	<code>punif(q, min, max)</code>	<code>runif(n, min, max)</code>
Gamma	<code>qgamma(p, s, r)</code>	<code>dgamma(x, s, r)</code>	<code>pgamma(q, s, r)</code>	<code>rgamma(n, s, r)</code>
F-Snedecora	<code>qf(p, df1, df2)</code>	<code>df(x, df1, df2)</code>	<code>pf(q, df1, df2)</code>	<code>rf(n, df1, df2)</code>
Dwumianowy	<code>qbinom(p, s, p)</code>	<code>dbinom(x, s, p)</code>	<code>pbinom(q, s, p)</code>	<code>rbinom(n, s, p)</code>
Poissona	<code>qpois(p, lambda)</code>	<code>dpois(x, lambda)</code>	<code>ppois(q, lambda)</code>	<code>rpois(n, lambda)</code>

Materiały do zajęć 1 z Programowania narzędzi analitycznych

1. Pomoc

`help("nazwa_polecenia")` - wyświetla informacje o poleceniu

`?nazwa_polecenia` - wyświetla informacje o poleceniu

`help.search("słowoKluczowe")` - Przemysław Biecek, Przewodnik po pakiecie R, 2008, str. 18, (link): Przegląda opisy funkcji znajdujących się w zainstalowanych pakietach i wyświetla te pozycje, w których znaleziono wskazane słowoKluczowe. W tym przypadku słowoKluczowe może oznaczać również kilka słów lub zwrot. W liście wyników znajduje się również informacja, w którym pakiecie znajdują się znalezione funkcje.

`demo` - pliki instruktażowe. Przemysław Biecek, Przewodnik po pakiecie R, 2008, str. 10, (link): Dla wielu pakietów oraz funkcji dostępnych w R zostały przygotowane prezentacje, pokazujące możliwości danego pakietu lub funkcji. Takie prezentacje uruchamia się funkcją `demo(utils)`.

2. Polecenia ogólne

`rm(a)` - usuwa zmienną `a`

`rm(list=ls())` - usuwa z pamięci komputera wszystkie zmienne i funkcje

`ctrl+l` - czyści okno poleceń

`getwd` - podaje bieżący katalog roboczy

`setwd` - zmiana katalogu domyślnego

`Sys.Time()` - wyświetla datę i godzinę

`Sys.Date()` - wyświetla datę

`q()` - wyłączenie programu R

`#` - tworzenie komentarza

3. Działania matematyczne

`=` lub `<-` - przypisanie wartości np.: `a=4`

`+`, `-`, `*`, `/` - podstawowe działania matematyczne

`^` - podniesienie do potęgi

4. Operatory i funkcje logiczne

`==` - równy, porównanie

`!=` - nierówny

`!` - nierówny, negacja

`<`, `<=`, `>`, `>=` - relacje

`&` - koniunkcja, "i", "AND"

`|` - alternatywa, "lub", "OR"

`xor` - alternatywa wykluczająca

`is.infinite(x)` - czy liczba `x` jest równa nieskończoności

`is.finite(x)` - czy liczba `x` jest skończona

`is.nan(x)` - czy `x` jest brakiem danych

5. Podstawowe funkcje

`sqrt(x)` - pierwiastek z liczby `x` (\sqrt{x})

`exp(x)` - eksponenta z `x` (e^x)

`log(x)` - logarytm naturalny ($\ln(x)$)

`log10(x)` - logarytm o podstawie 10 z liczby `x`
`log2(x)` - logarytm o podstawie 2 z liczby `x`
`log(x, base=b)` - logarytm o podstawie `b` z liczby `x`
`sin()`, `cos()`, `tan()` - funkcje trygonometryczne
`asin()`, `acos()`, `atan()` - odwrotne funkcje trygonometryczne
`sinh()`, `cosh()`, `tanh()` - funkcje hiperboliczne
`abs()` - wartość bezwzględna
`factorial(n)` - silnia z liczby `n` ($n!$)
`floor()` - zaokrąglenie liczby w dół
`ceiling()` - zaokrąglenie liczby w górę
`c %% d` - reszta z dzielenia liczby `c` przez liczbę `d`, modulo
`round()` - zaokrąglenie do najbliższej liczby całkowitej
`sign()` - zwraca (-1) dla liczb ujemnych i 1 dla dodatnich i 0 dla zera

6. Stałe

`pi` = π = 3.14159
`Inf` - nieskończoność
`NaN` - brak danych

7. Generowanie macierzy i wektorów

`matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE)` - tworzy macierz
`diag(wielkość)` - tworzy macierz jednostkową
`diag(macierz)` - wybiera diagonalę z macierzy

8. Funkcje dla macierzy i wektorów

`det(macierz)` - wyznacznik macierzy
`t(A)` - zwraca macierz transponowaną do macierzy `A`
`length(wektor)` - zwraca długość wektora
`dim(macierz)` - zwraca wektor z wymiarami macierzy
`nrow(macierz)` - zwraca liczbę wierszy macierzy
`ncol(macierz)` - zwraca liczbę kolumn macierzy
`eigen(macierz)` - zwraca wartości i/lub wektory własne macierzy
`sum(wektor)` - suma elementów wektora lub macierzy
`min(wektor)` - wyznacza minimalną wartość z wektora
`max(wektor)` - wyznacza maksymalną wartość z wektora
`prod(wektor)` - wyznacza iloczyn wszystkich elementów wektora
`cumsum(wektor)` - wylicza sumę narastającą
`mean(wektor)` - wylicza średnią z wektora
`median(wektor)` - wyznacza medianę z wektora
`sd(wektor)` - wylicza odchylenie standardowe z wartości wektora
`which` - funkcja znajduje elementy macierzy spełniające warunek

10. Łączenie macierzy i wektorów

`c(v1, v2)` - łączenie wektorów `v1` i `v2`
`rbind(v1, v2)` - łączenie wektorów `v1` i `v2` wiersz pod wierszem
`cbind(v1, v2)` - łączenie wektorów `v1` i `v2` kolumna obok kolumny