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 kwartyl

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 dystrybuanty rozkładu w punkcie `x`, tj. $\Phi(x)$

Rozkład	Kwantyl	Gęstość	Dystrybuanta	Liczby losowe
Normalny	**q**norm(p, mean, sd)	**d**norm(x,mean, sd)	**p**norm(q, mean, sd)	**r**norm(n, mean, sd)
Beta	**q**beta(p, s1, s2)	**d**beta(x, s1, s2)	**p**beta(q, s1, s2)	**r**beta(n, s1, s2)
χ^2_n	**q**chisq(p, df)	**d**chisq(x, df)	**p**chisq(q, df)	**r**chisq(n, df)
Wykładniczy	**q**exp(p, rate)	**d**exp(x, rate)	**p**exp(q, rate)	**r**exp(n, rate)
t-Studenta	**q**t(p, df)	**d**t(x, df)	**p**t(q, df)	**r**t(n, df)
Jednostajny(0,1)	**q**unif(p, min, max)	**d**unif(x, min, max)	**p**unif(q, min, max)	**r**unif(n, min, max)
Gamma	**q**gamma(p, s, r)	**d**gamma(x, s, r)	**p**gamma(q, s, r)	**r**gamma(n, s, r)
F-Snedecora	**q**f(p, df1, df2)	**d**f(x, df1, df2)	**p**f(q, df1, df2)	**r**f(n, df1, df2)
Dwumianowy	**q**binom(p, s, p)	**d**binom(x, s, p)	**p**binom(q, s, p)	**r**binom(n, s, p)
Poissona	**q**pois(p, lambda)	**d**pois(x, lambda)	**p**pois(q, lambda)	**r**pois(n, lambda)

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(''slowoKluczowe'')` - 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 slowoKluczowe. W tym przypadku slowoKluczowe 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` $= \pi = 3.14159$
`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