Úvod do programu Matlab

MATLAB je špičkové integrované prostředí pro vědeckotechnické výpočty, modelování, simulace, prezentaci a analýzu dat. Je to nástroj jak pro pohodlnou interaktivní práci, tak pro vývoj širokého spektra aplikací. Poskytuje svým uživatelům mocné grafické a výpočetní nástroje, ale i rozsáhlé knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace.

Za nejsilnější stránku tohoto programu je považováno mimořádně rychlé výpočetní jádro s optimálními algoritmy (programované ve Fortranu a jazyce C), které jsou prověřeny léty provozu na špičkových pracovištích po celém světě. Jeho implementace je provedena na všech významnějších platformách, od osobních počítačů s operačními systémy Windows 95, Windows NT a Linux přes počítače Macintosh až po pracovní stanice Sun, Hewlett-Packard, IBM nebo DEC a grafické stanice Silicon Graphics.

Výrobce programu MATLAB je firma The Maths Works - http://www.mathworks.com.
Českým distributorem je firma Humusoft s.r.o. - http://www.humusoft.cz.

Základní vlastnosti:

- Interpretační jazyk - uživatel obdrží odpověď na svůj povel téměř okamžitě.
- Základními objekty - matice, ale podporuje i složitější typy, například vícerozměrná pole, datové struktury,….
- Skládáním datových typů je možné vytvořit libovolně složité datové struktury.
- Grafika - umožňuje snadné zobrazení a prezentaci získaných výsledků. Lze vykreslit různé druhy grafů: dvourozměrné, třírozměrné, histogramy, apod. MATLAB také umožňuje otevřít více oken pro zobrazení grafů najednou nebo zobrazit více grafů v jednom okně.
- Otevřená architektura přispěla k velkému rozšíření.
- Úplný programovací jazyk - uživatelé v něm mohou vytvářet funkce "šité na míru" pro jejich aplikace. Tyto funkce se způsobem volání nijak neliší od vestavěných funkcí a jsou uloženy v souborech v čitelné formě. Dokonce většina funkcí s MATLABem dodávaných je takto vytvořena a opravdu vestavěné jsou jen funkce základní. To má dvě velké výhody: jazyk MATLABu je téměř neomezeně rozšiřitelný a kromě toho se uživatel může při psaní vlastních funkcí poučit z algoritmů s programem dodávaných. Navíc jsou takto koncipované funkce snadno přenosné mezi různými platformami, na kterých je MATLAB implementován.

1 Charakteristika datových struktur

- speciální znaky

%	poznámka
;	konec příkazu s potlačením výstupu výsledku
	konec řádky v matici
!	uvádí běžný příkaz operačního systému
…	pokračování příkazu na dalším řádku
( )	závorky výrazů, indexové závorky
[ ]	maticové závorky
{ }	buňky

- základní datové typy (MATLAB rozlišuje velká a malá písmena v názvech proměnných, nerozlišuje malá a velká písmena v názvech funkcí)

typ	příklad	velikost
číslo	5	matice 1 x 1, reálné číslo typu double, int8, int16, int32, uint8 (bezznaménkové int8), uint16, uint32
	1.2e+3	
	2e-2	
	-1/5	
	5+i*3	matice 1 x1 , komplexní číslo
znak	'a'	matice 1 x 1
	char(64)	
řetězec	'toto bude text'	matice 1 x počet znaků textu
datum	now	aktuální datum a čas jako číslo
	date	aktuální datum jako řetězec : dd-mmm-yyyy
	clock	aktuální datum jako vektor čísel [y m d h m s]
matice	[1 2 3 4 5 6]	matice řádek x sloupec matice 2 x 3
	[1,2,3;4,5,6]	
	[1 2 3;4 5 6]	
	['ahoj';'bye ']	matice 2 x 4
	['ahoj','bye ']	matice 1x 8
	[1'a';22 'b']	matice 2 x 2
	sparse(X)	řídká matice X
vícerozměrné pole	a=[1,2,3;4,5,6]	matice řádek matice a x sloupec matice a x 2
	b=[7,8,9;10,11,12]	c(i,j,1)=a(i,j)
	c=cat(3,a,b)	c(i,j,2)=b(i,j)

buňka	c=cell(1) c{1}	vytvoří jednu prázdnou buňku, která může obsahovat číslo, znak, řetězec, matici, vicerozměrné pole, buňku,…
pole buněk	c=cell(M,N,P,…) c{m,n,p,…}	vytvoří MxNxPx… rozměrné pole prázdných buněk, která mohou obsahovat číslo, znak, řetězec, matici, vicerozměrné pole, buňku,…
strukturální pole	s = struct('type',{'big','little'},'color',{'red'},'x',{3 4}) s(1)= type: 'little' color: 'red' x: 4	
inline funkce	f=inline('t^2') g=inline('sin(1/x)') f(2) g(f(pi))	funkce $f(t)=t^2$ funkce $g(x)=\sin(1/x)$ 2^2 $\sin(1/pi*pi)$
objekty		

- konverze jednotlivých datových typů

  Existují předefinované funkce na konverzi jednotlivých proměnných, zadaných typů parametrů vybraných úloh apod.

  - konverze mezi jednotlivými typy čísel;
  - konverze mezi řetězci a čísly;
  - konverze datumových proměnných;
  - konverze buněk na strukturální pole;
  - konverze numerických hodnot matice na buňky;
  - konverze různých souřadnic (kartézské, sférické, …)
  - ….

  - důležitý je příkaz *eval (řetězec)*, který přečte řetězec a provede ho jako příkaz programu MATLAB

  Příklad: (vytvořit 10 pseudonáhodných matici N x N označených M1,M2,…M10)

```
N=5
for index =1 : 10
    matice_s_indexem= ['M',int2str(index),' = rand(N)'];
    eval(matice_s_indexem)
end
```

2 Vstup a výstup dat

vstup dat

- interaktivně
  v rámci pracovního okna lze zadávat libovolná data pomocí přiřazovacího příkazu =
- po výzvě z klávesnice
  pomocí příkazu *proměnná=input('textová výzva')*, program zobrazí "textovou výzvu" a čeká na vložení dat z klávesnice, data přiřadí *proměnné*
- v MAT – souboru (standardní přípona souboru je .mat, ale lze použít libovolnou)
  - proměnné lze pomocí příkazu *save jméno_souboru (případně s cestou)* uložit do souboru a později použít pomocí příkazu *load jméno_souboru (případně s cestou) jméno_proměnných*
  - soubor příkazu *load* může být ASCII soubor s daty po řádkách oddělenými mezerami, MATLAB načte data do proměnné stejného jména jako je ASCII soubor
- v M-souboru
  pokud jsou data součástí přiřazovacích příkazů v programu
- načtením TXT souboru pomocí příkazů *fopen, fscanf, fread,…textread* (pro MATLAB 6)
- načtením WK1 souboru pomocí příkazů *wk1read*
- tabulkový soubor – pomocí *tblread, tdfread, caseread*
- načtením Excel souboru (nutno dokoupit toolbox Excel LINK)
- vnitřní paměťové proměnné
  - *eps* (2.10^{-16}),
  - *pi* (3,14…),
  - *Inf*
  - *NaN* (Not a Number),
  - *i, j* (imaginární jednička),
  - *ans* (jméno výsledku nepřiřazeného výrazu)
  - *realmax, realmin* maximální a minimální reálné číslo (1.7977e+308, 2.2251e-308)

výstup dat

- paměťové proměnné – přiřazovacími příkazy
- interaktivně text – pokud M-příkaz nekončí ; je automaticky vypsán výsledek do pracovního okna
- MAT soubor – pomocí příkazu *save*
- TXT soubor – pomocí příkazů *fwrite, sprintf, …*
- WK1 soubor – pomocí příkazů *wk1write*
- tabulkový soubor – pomocí *tblwrite, casewrite*
- Excel soubor (nutno dokoupit toolbox Excel LINK)
- grafy 2D

- vytvoření grafu *plot(X,Y), plot(X,Y,S)*, kde S je tříznakový řetězec charakterizující barvu, typ bodu a typ čáry př.'c+:' (c cyan, + typ bodu plus, : typ čáry dotted)
- *plot(X_1,Y_1,S_1,X_2,Y_2,S_2,…)* více funkcí do jednoho grafu
- *loglog(), semilogx(),semilogy()* logaritmické měřítko os
- *polar()* polární souřadnice
- *area(X,Y)* plocha pod křivkou
- *bar(), barh(), bar3h()* sloupcový graf a sloupcový horizontální graf
- *comet()* animovaný průběh grafu
- *errorbar(X,Y,ERR)*
- *ezplot('funkce'), fplot('funkce',argument)* jednoduché vykreslení funkce zadané jako řetězec nebo jako inline funkce
- *pie(), pie3()* koláčové grafy
- *plotmatrix(X,Y)* vykreslí matici NxM grafů zachycující závislost i-tého sloupce matice X (i=1,…,N) a j-tého sloupce matice Y (j=1,…,M)
- *ribbon(X,Y)* vykreslí funkci jako pásku
- *stem(X,Y)* diskrétní posloupnost bodů
- *stairs(X,Y)* "schodovitá" funkce

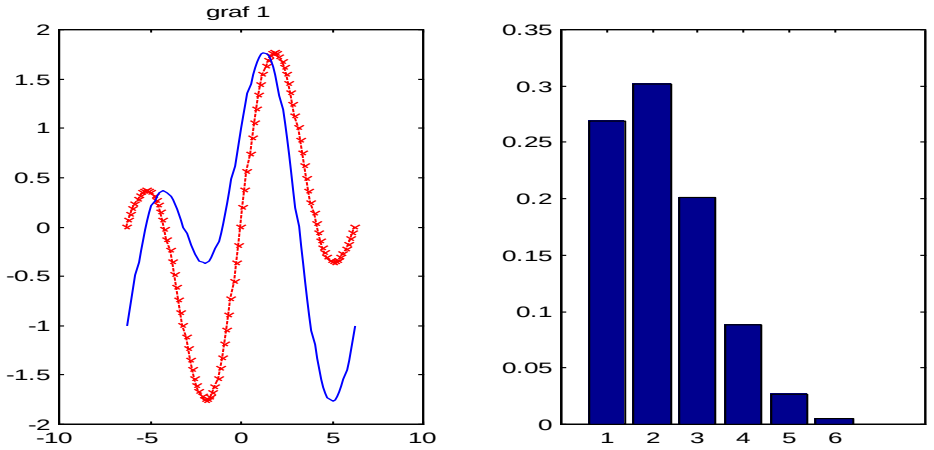
- úprava grafů: *title(), xlabel(), ylabel(), text(), gtext(), grid, …*

- grafy 3D
  - *plot3(x,y,Z)*
  - *[X,Y]=meshgrid(x,y)* vvytvoření sítě z vektorů souřadnic x a y
  - *mesh(Z)* síťový graf 3D
  - *surf(Z)* plošný graf 3D
  - *surfl(Z)* plošný graf 3D
  - *surfc(Z)* plošný graf 3D s vrstevnicemi
  - *ribbon(Y,Z)* páskový graf
  - *contour(Z)* vrstevnicový graf
  - *fill3(X,Y,Z)* vyplněný graf
- úprava grafů: *title(), xlabel(), ylabel(), text(), gtext(), grid, shading, colormap(), …*
- grafy standardně vystupují do jednoho okna, lze kopírovat do jiných SW (například Microsoft), ukládat jako M-soubor,…
- výstupní okno lze rozdělit na několik oblastí pomocí příkazu *subplot*
- vytvoření nového okna je realizováno příkazem *figure*, zachování daného okna příkazem *hold on*

Příklad vykreslení grafu 2D funkce

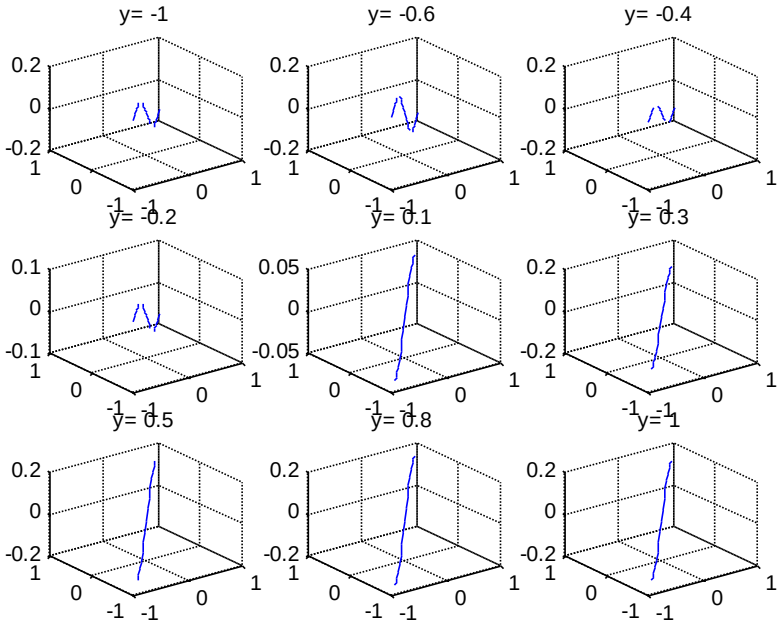
```
subplot(1,2,1)
x=-2*pi:4*pi/100:2*pi;
plot(x,sin(x)+cos(x/2))
plot(x,sin(x/2)+sin(x),'rx--')
hold on
plot(x,sin(x)+cos(x/2))
title('graf 1')
```

```
subplot(1,2,2)
a=1:6;
b=binopdf(a,10,.2);
bar(a,b)
```

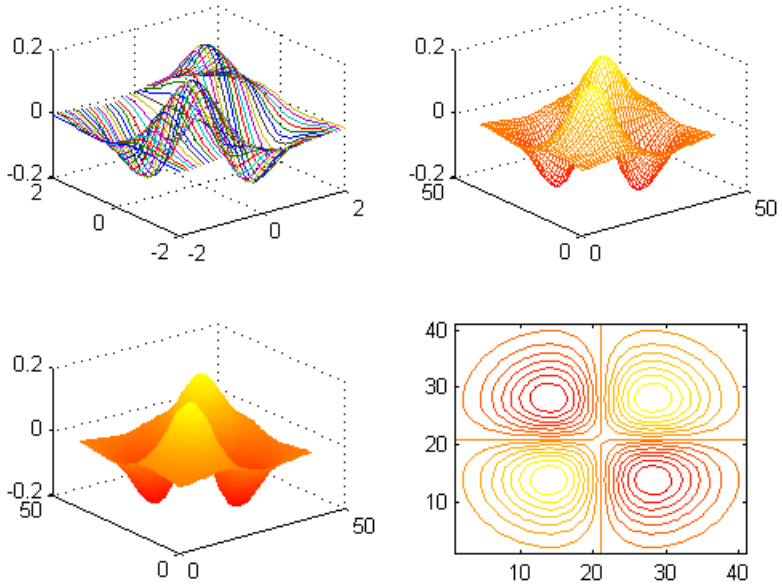


- Příklad vykreslení grafu 3D funkce

```
close all
clear all
x=-1:.1:1;
y=-1:.1:1;
f=inline('x.*exp(-x.^2)*y.*exp(-y.^2)','x','y');
n=9;
for k=1:n,
  if k==1,a(k)=1;
  else a(k)=round((k*length(x)/(n)));
  end
  subplot(3,3,k)
  plot3(x,y,f(x,y(a(k))))
  grid
  title (['y= 'num2str(y(a(k)))])
end
```



- Příklad vykreslení grafu 3D funkce



```
close all
clear all
[x,y]=meshgrid(-2:.1:2,-2:.1:2);
f2=inline('x.*exp(-x.^2).*y.*exp(-y.^2)','x','y');
z2=f2(x,y);
set(figure,'Name',['Funkce f2= 'formula(f2)],'NumberTitle','Off')
shading flat
colormap(autumn)
subplot(2,2,1),plot3(x,y,z2),grid
```

```
subplot(2,2,2),mesh(z2),rotate3d on,
subplot(2,2,3),surf(z2),shading interp
subplot(2,2,4),contour(z2,15)
```

3 Seznam základních příkazů a funkcí

- generování vektorů *(vektor=zacatek:krok:konec, linspace(), logspace(),…)*
  - *1:10* generuje 1,2,3,…10
  - *0:.1:10* generuje 0,0.1,0.2,…,9.9,10
  - *(0:10)^2* generuje 0,1,4,9,16,…100

- práce s maticemi
  - *A(i,j)* – (i,j)-tý prvek matice A
  - *A(i,:)* – i-tá řádka matice A
  - *A(:,j)* – j-tý sloupec matice A
  - *A(:*) – celá matice jako vektor sloupcových vektorů
  - *A(k)* – *k*-tý prvek matice A representované jako vektor sloupcových vektorů
  - *A([1 3 5],[2 4])*– submatice matice A, která obsahuje 1,3 a 5 řádek a 2 a 4 sloupec
  - *A(:,n:-1:1)* – matice A s převráceným pořadím sloupců (n je počet sloupců)
  - *A(:,sum(A)>0)* – submatice matice A, která obsahuje pouze sloupce, jejichž suma je kladná
  - *A(sum(A,2)>=0,:)* – submatice matice A, která obsahuje pouze řádky, jejichž suma je nezáporná

- operátory
  - aritmetické operátory *plus(+), minus(-) mtimes(*) mrdivide(/) mldivide(\) mpower(^)*
  - prvkové operátory *times(.*) rdivide(./) ldivide(.\) power(.^)*
    - ✓ nechť A, B jsou matice typu 2 x 3, pak A/B = A*pinv(B) je typu 2x2, $A\backslash B=(B^T/A^T)^T$ je typu 3 x 3, $A./B=[a_{ij}/b_{ij}]_{ij}$ je typu 2 x 3 a $A.\backslash B=[b_{ij}/a_{ij}]_{ij}$ je typu 2 x 3
    - ✓ nechť A je čtvercová matice , pak A^2=A*A, A^(1/2)=V*D^(1/2)*inv(V) kde V je matice vlastních vektorů a D je diagonální matice s vlastními čísly na diagonále
    - ✓ nechť A je libovolná matice, pak $A.\hat{}p=[a_{ij}\hat{}p]_{ij}$
  - relační a logické operátory a funkce *eg(==), ne(~=), lt(<), gt(>), le(<=), ge(>=), and(&), or(!), not(~) xor, any(A,dim), all(A,dim), isfinite(A), isnan(A), isinf(A), find(logický výraz)*
    - ✓ ~(A==B) prvek je jedna, když $a_{ij}<>b_{ij}$, jinak je prvek nula
    - ✓ x(x>0) vypustí z vektoru x nekladné prvky a zbytek přeindexuje
    - ✓ A(A>0) vypustí z matice A nekladné prvky a zbytek přeidexuje jako vektor sloupcových vektorů
    - ✓ *any(A,1)* vrátí vektor délky počet sloupců, prvek je jednička, když existuje alespoň jeden nenulový prvek ve sloupci, jinak je prvek nula
    - ✓ *all(A,2)* vrátí vektor délky počet řádků, prvek je jednička, když všechny prvky v řádku jsou nenulové, jinak je prvek nula

- ✓ *find(A>0)* vrátí vektor indexů, kde pro které a_{ij}>0 (počítáno jako vektor sloupcových vektorů)

- programové příkazy
  - řídící a informační příkazy *(clear, quit, exit, help, help demo, demo, who, whos, return, pause, pack, exist, close …)*
  - podmíněné příkazy
    - ✓ *if logický výraz odd příkaz odd elseif logický výraz odd příkaz odd … else odd příkaz end* (kde odd je ENTER nebo čárka nebo středník)

```
if       k<0, a=-1;
  elseif k==0, a=0;
  elseif k>0,a=1;
  else a=NaN;
end
```

```
if k<0, a=-1,elseif k==0, a=0,elseif k>0,a=1,else a=NaN,end
```

    - ✓ *switch switch_výraz, case case_výraz, příkaz, case case_výraz, příkaz,…, otherwise, příkaz, end*

  - příkazy cyklu
    - ✓ *for proměnná=M-výraz, příkaz, příkaz, … , end*
      - ❖ pokud M-výraz je skalár , cyklus proběhne jednou
      - ❖ pokud M-výraz je vektor *v* délky k , cyklus proběhne k-krát, proměnná nabývá postupně hodnot v(i), i=1,2,…k
      - ❖ pokud M-výraz je matice A typu m x n , cyklus proběhne n-krát, proměnná nabývá postupně hodnot A(:,j), kde j=1,2,…n
      - ❖ pokud M-výraz je matice A typu m x n x p, cyklus proběhne (n x p)- krát, proměnná nabývá postupně hodnot A(:,j,k), kde j=1,2,…n a k=1,2,..p
      - ❖ přerušení cyklu lze realizovat pomocí *break*
  - *while M-výraz, příkaz, příkaz,…, end*
    - ❖ přerušení cyklu lze realizovat pomocí *break*

  - makrokříkazy

    - ✓ funkce – úsek programu uložený jako M-soubor, vstupní argumenty vyhodnotí jako parametry funkce, všechny proměnné jsou lokální, shoda jmen s globálními proměnnými nevadí, vyvolává se pomocí názvu M-souboru

      Příklad funkce pro výpočet základních statistických ukazatelů, uloženo jako zakl_stat.m

```
function [prumer,rozptyl,vyb_rozptyl,hlaska]=zakl_stat(data,nazev_dat)
```

```
%Vypocet zakladnich statistickych charakteristik
%Povinny parametr jsou data, nesmi byt nulovy
prumer=NaN;
rozptyl=NaN;
vyb_rozptyl=NaN;
if nargin<2
  hlaska='"nezadany nazev" ';
elseif nargin<1
  hlaska='je treba zadat parametry data a nazev dat';
  return;
else
  if isstr(nazev_dat),hlaska=nazev_dat;
  else hlaska='"chybny nazev"';
  end
end
if isstr(data), hlaska=[hlaska '(TXT DATA}'];end
data=data(:);
n=length(data);
if n<1
  hlaska=['CHYBA >> velikost ' hlaska ' = 'int2str(n)];
elseif n==1
  prumer=sum(data)/n;
  rozptyl=sum((data-prumer).^2)/n;
  vyb_rozptyl=rozptyl;
  hlaska=['velikost  ' hlaska '=  ' int2str(n)];

else
  prumer=sum(data)/n;
  rozptyl=sum((data-prumer).^2)/n;
  vyb_rozptyl=rozptyl*n/(n-1);
  hlaska=['velikost 'hlaska ' = ' int2str(n)];
end
```

- skripty – úsek programu uložený jako M-soubor pod daným jménem, nemá argumenty, definované proměnné jsou globální, skript se při každém vyvolání znovu překládá
- *eval(řetězec)* funkce MATLABu, které vyhodnotí *řetězec* jako M-příkaz a provede ho (viz konverze jednotlivých datových typů)
- *feval(řetězec, x1,x2,…)* vyvolá funkci uloženou v M-souboru pod názvem *řetězec* s parametry *x1,x2,…*

- matematické funkce
    - trigonometrické fce *sin, cos, …*
    - exponenciální fce *exp, log,…*
    - komplexní fce *abs,angle,…*
    - zaokrouhlování *round*, *fix, floor,…*
    - speciální funkce *beta,gamma, besselj,…*
    - funkce kombinatoriky *faktor, perms,…*
    - transformace souřadnic *cart2sph, cart2pol,…*

- maticové funkce
    - generování speciálních matic *zeros, ones, eye, rand, magik, hilbert, pascal,…*
    - operace s maticemi *diag, expm, logm, sgrtm, funm (př. funm(A,'sin')),*
    - transformace matic A'
    - charakteristiky matice *size, trace, det, norm, rank, cond,…*
    - inverze matic *inv, pinv*
    - rozklady matic *lu, qr, chol, svd, …* a báze *orth, null,…*
    - problémy vlastních čísel a vlastních vektorů *eig, …*
    - funkce pro operace s řídkými maticemi

- funkce matematické analýzy
    - polynomy *poly, roots, conv,…*
    - interpolace *spline, table1,…*
    - derivace a integrace *diff, gradinet, quad,…*
    - kořeny fce *roots, zeroin,…*
    - řešení diferenciálních rovnic *ode23,…*
    - Fourierova transformace
    - minimalizace *max, min, fmin, fmax,…*
    - …

- optimalizační funkce (část těchto funkcí je součástí modulu Optimization Toolbox)
    - volné extrémy – *fmin, fmins, fminu*
    - vázané extrémy – lineární programování, kvadratické programování, obecné funkce pro řešení vázaných extrémů – *lp, qp, constr*
    - speciální typy minimalizací – *seminf, attgoal*
    - minmax optimalizace – *minimax*
    - řešení rovnic rovnic f(x)=0 – *fzero, fsolve, \,*
    - vyrovnávání křivek (MNČ) – *curvefit, leastsq, nnls, conls, *

- statistické funkce - deskritptivní statistiky
    - maximum, minimum – *max, min*
    - průměry – *mean, geomean, hermmean, trimmean*
    - rozptyly, směrodatné odchylky – *var,std*

- šikmost, špičatost, momenty – *kurtosis, skewness, moment*
- robusní statistiky – *iqr, mad, median, range*
- kvartily, percentily - *quantile, prctile*
- celkové statistiky – *grpstats*
- vícerozměrné statistiky – *corr, cov*

- různé typy rozdělení
  - pdf – funkce hustoty,
  - cdf – distribuční funkce,
  - inv – kvantilové funkce,
  - fit – odhady parametrů
  - rnd – generátory náhodných čísel,
  - stat – momenty

  - *normpdf, normcdf, norminv,Normand, normstat*
  - *beta rozdělení, negativní beta rozdělení, rozdělení extrémních hodnot, exponenciální, binomické, chi-kvadrat, F-rozdělení, Gamma rozdělení, Studentovo, Weibullovo, logaritmicko normální, rovnoměrné ,….*

- Statistické grafy
  - *boxplot, hist, normplot, qqplot, wblplot, errorbar, pareto*

- Statistické hypotézy
  - Parametrické
    - ✓ *ttest, ttest2,ztest, anova*
  - Neparametrické
    - ✓ *signrank, ranksum,signtest, friidman, kruskalwallis, kstest,lillietest,jbtest*