

25. Matlab - Import a Export textových súborov

Blaho Michal · MATLAB/Comsol

18.06.2010



V [poslednej časti](#) seriálu o Matlabe sme sa venovali importu a exportu do štandardných dátových súborov Matlabu MAT-súborov. V dnešnej časti si ukážeme prácu s najčastejšie používanými typmi súborov, ktoré označujeme ako textové.

Import textových súborov

Na import textových dát z príkazového riadka alebo M-súboru musíme zavolať niektorú z Matlabových funkcií pre import. Výber funkcie zväčša závisí od toho, ako sú dáta v súbore naformátované. Textové dáta musia byť naformátované do rovnakého počtu riadkov a stĺpcov. Ako oddeľovač dát slúži oddeľovací znak. Oddeľovací znak môže byť medzera, čiarka, tabulátor alebo iný znak. Dáta môžu byť v súbore číselné prípadne znakové. Niektoré súbory môžu obsahovať tzv. hlavičky ku dátam na ich označenie.

Ak sú dáta uložené v súbore do obdĺžnika (rovnaký počet stĺpcov v riadku) najjednoduchším spôsobom ako ich načítať je s využitím funkcie **load**. Jednotlivé prvky by mali byť oddelené medzerou. Dáta sú importované do Workspace s rovnakým menom ako je meno súboru bez koncovky. Ak chceme určiť meno pod ktorým sa dáta uložia musíme príkaz použiť ako v príklade na druhom riadku.

```
>>load data.txt;
>>A = load('data.txt');
```

Pre iné oddeľovače ako je medzera môžeme využiť pokročilejšie príkazy ako je napríklad **dlmread**. Ako druhý parameter tejto funkcie uvádzame oddeľovač ako znak. Príkaz ignoruje medzery medzi dátami. Ako ďalšie parametre môžeme uviesť posunutie dát, od ktorých sa má začať čítať (indexuje sa od nuly).

```
>>A = dlmread('data.txt', ';');
>>A = dlmread('data.txt', ';', 2, 3);
```

Podobný príkaz ako dlmread je príkaz **csvread**. Tento príkaz sa využíva na čítanie dát z textových procesorov uložených v csv súboroch. V týchto súboroch sa dáta oddeľujú čiarkou preto tento znak nemusíme zadávať ako v príkaze dlmread. Zvyšné parametre zostávajú.

```
>>A=csvread('data.txt', 2, 3);
```

Niekedy textový súbor obsahuje hlavičky ku dátam, ktoré treba často vynechať. V takomto prípade využijeme funkciu **textscan**. V tejto funkcii môžeme zadať parameter **headerlines**, ktorý slúži na ignorovanie prvých N riadkov (kde sa nachádzajú hlavičky). Použitie si ukážame na nasledovnom súbore, ktorý obsahuje nasledujúce dáta.

```
Grade1 Grade2 Grade3
78.8 55.9 45.9
99.5 66.8 78.0
89.5 77.0 56.7
```

Dáta zo súboru *data.txt* môžeme načítať nasledovne:

```
>>fid = fopen('data.txt', 'r');
>>grades = textscan(fid, '%f %f %f', 3, 'headerlines', 1);
>>fclose(fid);
```

V predchádzajúcom príklade sme si najskôr museli otvoriť súbor na čítanie. Premenná *fid* obsahuje identifikátor súboru. Príkazom **textscan** sme načítali 3 riadky pričom dáta boli načítané ako čísla s dvojitou presnosťou a vynechali sme prvý riadok, kde sú hlavičky. Znak `%f` je typ dát a sú totožné so znakmi konverzie, ktoré sme si už popisovali v článku o [formatovom výstupe](#) v tabuľke 7 prípadne v manuály k príkazu. Súbor musí byť uzavretý príkazom **fclose**.

Prepínače pre určenie typu dát zohrávajú dôležitú úlohu pri čítaní textových súborov, ktoré majú v riadku rôzny typ dát. Tieto dáta môžeme čítať pomocou funkcií **textscan** a **textread**. Príkaz **textscan** sme si už naznačili a ešte dodáme, že jeho výstup sa ukladá do premennej v ktorej je pole typu `cell`. Príkaz **textread** dokáže uložiť dáta do nami špecifikovaných premenných. Funkcia **textscan** je rýchlejšia pri väčších súboroch. Použitie si ukážeme na nasledujúcich dátach, ktoré nájdete aj v užívateľskej príručke.

```
Sally Type1 12.34 45 Yes
Larry Type2 34.56 54 Yes
Tommy Type1 67.89 23 No
```

Takéto dáta načítame pomocou príkazu **textread** nasledovne

```
>>[names, types, x, y, answer] = ...
textread('data.txt', '%s %s %f %d %s', 3)
```

Príkaz rozdelí dáta v súbore do jednotlivých premenných pričom ich typy určujú prepínače a počet riadkov na spravovanie sa rovná poslednému parametru.

Export do textových súborov

Pre export dát do textových súborov má Matlab vstavaných niekoľko funkcií. Výber funkcie záleží od množstva dát a formátu výstupného súboru. Už v minulej časti sme si

popísali najjednoduchšiu funkciu **save**. Funkcia slúži na ukladanie do binárnych MAT-súborov, ale aj na ukladanie do ASCII súborov a ako oddeľovač používa medzeru. Nasledujúce príkazy by vám už nemali byť neznáme.

```
>>A = [1 2 3 4 ; 5 6 7 8];  
>>save data.out A -ASCII
```

Pre pokročilejšie možnosti exportu môžete využiť príkaz **dlmwrite**. Príkaz umožňuje špecifikovať vlastný oddeľovač dát, výber len určitej časti dát z matice alebo vektora prípadne znak ukončenia riadku, ktoré sa líšia pre Windowsové a Unixové systémy. Príkaz nepridáva oddeľovač na konci riadku. Oddelovač sa pridáva do príkazu ako znak (teda v úvodzovkách) pričom sa štandardne využíva čiarka. Začiatok dát v matici sa zadáva číslom riadku a stĺpca pričom index prvého prvku je 0,0.

```
>>A = [1 2 3 4 ; 5 6 7 8];  
>>dlmwrite('data.out',A, ';' )  
>>dlmwrite('data.out',A, ';' , 1, 2)
```

Podobný príkaz ako dlmwrite je príkaz **csvwrite**. Tento príkaz je však primárne určený pre dáta, ktoré sa využívajú v tabuľkových procesoroch. Na rozdiel od príkazu dlmwrite nešpecifikujeme oddeľovač, ktorý je vždy čiarka.

```
>>A = [1 2 3 4 ; 5 6 7 8];  
>>csvwrite('csvdata.out',A, 1, 2)
```

Posledným príkazom na ukladanie dát do textového súboru je príkaz **diary**. Tento príkaz však ukladá výstup z príkazového riadku Matlabu.

```
>>diary data.out  
>>A = [1 2 3 4 ; 5 6 7 8];  
>>A  
>>diary off  
>>type data.out
```

Posledným príkazom vypíšeme obsah súboru data.out do príkazového riadku. Týmto príkazom ukončíme dnešný diel seriálu o Matlabe a v nasledujúcej časti sa pozrieme na import a export z tabuľkových procesorov. Pokročilé použitie uvedených príkazov nájdete v užívateľskej príručke spolu s ich ďalšími možnosťami a parametrami.