Working With Data

High Level I/O

- save: export workspace variables to file
- load: import file into workspace variables
- fileread: read an entire file as text
- xlsread: load an Excel file
- xlswrite: write an Excel file
- importdata: typically used to load a text file. loads into a struct with matrix and text data as separate fields.
- readtable: load an Excel or text file (loads into special **table** object)

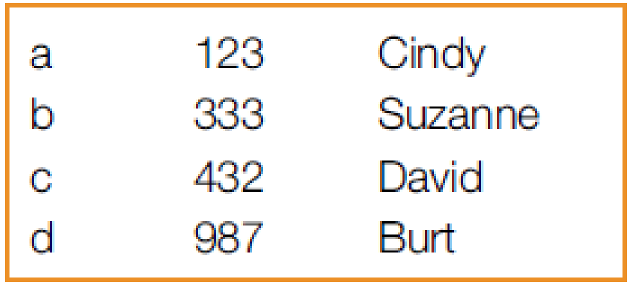
Working with Excel Files

```
>> a=randi(100,3,6);
>> xlswrite('rand.xlsx', a)
>> b=xlsread('rand.xlsx')

b =

    23    33    24    58    56    84
    19    80    69   100    93    18
    68    55     5    37    26    88
```

- WARNING: xlswrite() does not clear the entire worksheet; if there's data beyond the area that is written, they will still be there. Your best bet is to remove the file before you use xlswrite().

Excel files with mixed content

```
>> xlswrite('temp.xlsx',{'a' 'b' 'c' 'd'; ...
 123 333 432 987; ...
'Cindy' 'Suzanne' 'David' 'Burt'}')
```



a	123	Cindy
b	333	Suzanne
c	432	David
d	987	Burt

```
>> [nums, txt, raw] = xlsread('temp.xlsx')
```

```
nums =

   123
   333
   432
   987
```

```
txt =

  4×3 cell array

    {'a'}    {0×0 char}    {'Cindy'  }
    {'b'}    {0×0 char}    {'Suzanne'}
    {'c'}    {0×0 char}    {'David'  }
    {'d'}    {0×0 char}    {'Burt'   }
```

```
raw =

  4×3 cell array

    {'a'}    {[123]}    {'Cindy'  }
    {'b'}    {[333]}    {'Suzanne'}
    {'c'}    {[432]}    {'David'  }
    {'d'}    {[987]}    {'Burt'   }
```

Working with Excel Files that have a Header Row

% Sample file available at: http://sacan.biomed.drexel.edu/ftp/bmeprog/crps_data.xlsx

% Exercise: Find average pain score for female persons.

```
[~,~, raw] = xlsread('crps.xls’)
```

table object (t) vs. cell matrix (a)

- Column names are kept and managed by the table object.
- t(..., ...) : creates a new table object from selected entries
- t{ ..., ... } : automatically collects the data into most useful data type (may not be what you want, so double-check)
- t{ ..., 'columnname' } : can index columns by their names.

- You are responsible for keeping column names as first row of a, or keeping them in a separate variable.
- a(..., ...) : creates a new cell matrix from selected entries
- a{ ..., ... } : extracts the indexed entries. You must decide to collect them into vector [ ], or cell array { }.
- Selection by column name is not available. You are responsible for identifying the correct column number for the column name you want to select.