

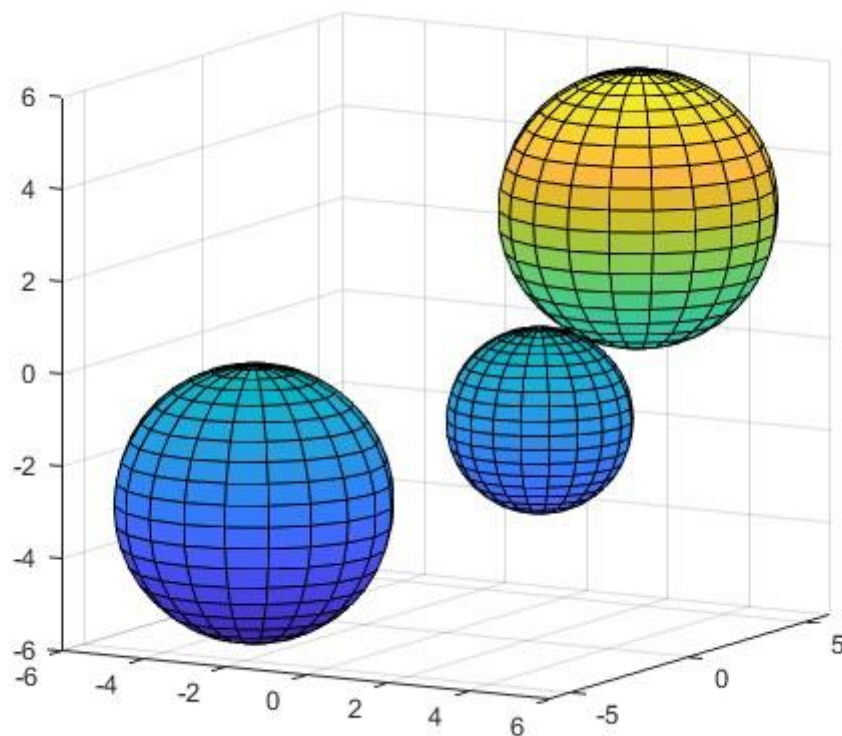
| | |
|--------------------------------|-----------------------|
| Roll No.: A016 | Name: Varun Khadayate |
| Class: B. Tech CsBs | Batch: 1 |
| Date of Experiment: 18-09-2022 | Subject: ITWS |

Lab 6: M-script file

1.
 - a. Create a file called mysphere.mat
 - b. Create and plot a sphere using sphere function. Sphere function creates a unit circle which can be scaled using surf function (hint: try different values of 'r' using surf function).
 - c. Run and observe the output
 - d. What is your observation?

While running the code with different value of r there are different sphere, and we can see that the dimension also changes. Below is the screenshot of 3 sphere in one graph

```
[x y z] = sphere;
a=[3 3 3 3;-3 -3 -3 3;0 4 -2 2];
s1=surf(x*a(1,4)+a(1,1),y*a(1,4)+a(1,2),z*a(1,4)+a(1,3));
hold on
s2=surf(x*a(2,4)+a(2,1),y*a(2,4)+a(2,2),z*a(2,4)+a(2,3));
s3=surf(x*a(3,4)+a(3,1),y*a(3,4)+a(3,2),z*a(3,4)+a(3,3));
daspect([1 1 1])
view(30,10)
```



2.

a. Create an excel sheet consisting the following data :

| Temperatures of Cities as on 20.6.2006 | | |
|---|------|------|
| City | Max. | Min. |
| Ahmedabad | 38°C | 29°C |
| Amritsar | 37°C | 26°C |
| Bangalore | 28°C | 21°C |
| Chennai | 36°C | 27°C |
| Delhi | 38°C | 28°C |
| Jaipur | 39°C | 29°C |
| Jammu | 41°C | 26°C |
| Mumbai | 32°C | 27°C |

b. Explore xlsread function using help

c. Read the excel file created in step a) using xlsread function

d. d) Create an M-File and write a script which finds the following:

i. Compare and identify the city that recorded the max temperature

ii. Plot the given data using bar graph

help **xlsread**

xlsread Read Microsoft Excel spreadsheet file.

*** **xlsread** is not recommended ***

See readtable, readmatrix, or readcell.

[NUM,TXT,RAW]=**xlsread**(FILE) reads data from the first worksheet in the Microsoft Excel spreadsheet file named FILE and returns the numeric data in array NUM. Optionally, returns the text fields in cell array TXT, and the unprocessed data (numbers and text) in cell array RAW.

[NUM,TXT,RAW]=**xlsread**(FILE,SHEET) reads the specified worksheet.

[NUM,TXT,RAW]=**xlsread**(FILE,SHEET,RANGE) reads from the specified SHEET and RANGE. Specify RANGE using the syntax 'C1:C2', where C1 and C2 are opposing corners of the region. Not supported for XLS files in BASIC mode.

[NUM,TXT,RAW]=**xlsread**(FILE,SHEET,RANGE,'basic') reads from the spreadsheet in BASIC mode, the default on systems without Excel for Windows. RANGE is supported for XLSX files only.

[NUM,TXT,RAW]=**xlsread**(FILE,RANGE) reads data from the specified RANGE of the first worksheet in the file. Not supported for XLS files in BASIC mode.

The following syntaxes are supported only on Windows systems with Excel software:

[NUM,TXT,RAW]=**xlsread**(FILE,-1) opens an Excel window to select data interactively.

[NUM,TXT,RAW,CUSTOM]=**xlsread**(FILE,SHEET,RANGE,'',FUNCTIONHANDLE) reads from the spreadsheet, executes the function associated with FUNCTIONHANDLE on the data, and returns the final results. Optionally, returns additional CUSTOM output, which is the second output from the function. **xlsread** does not change the data stored in the spreadsheet.

Input Arguments:

FILE Name of the file to read. **SHEET** Worksheet to read. One of the following:

- * The worksheet name.
- * Positive, integer-valued scalar indicating the worksheet index.

RANGE Character vector or string that specifies a rectangular portion of the worksheet to read. Not case sensitive. Use Excel A1 reference style. If you do not specify a **SHEET**, **RANGE** must include both corners and a colon character (:), even for a single cell (such as 'D2:D2').

'basic' Flag to request reading in BASIC mode, which is the default for systems without Excel for Windows. In BASIC mode, **xlsread**:

- * Reads XLS, XLSX, XLSM, XLTX, and XLTM files only.
- * Does not support an xlRange input when reading XLS files. In this case, use '' in place of xlRange.
- * For XLS files, requires a name to specify the **SHEET**, and the name is case sensitive.
- * Does not support function handle inputs.
- * Imports all dates as Excel serial date numbers. Excel serial date numbers use a different reference date than MATLAB date numbers.

-1 Flag to open an interactive Excel window for selecting data. Select the worksheet, drag and drop the mouse over the range you want, and click OK. Supported only on Windows systems with Excel software.

FUNCTIONHANDLE

Handle to your custom function. When **xlsread** calls your function, it passes a range interface from Excel to provide access to the data. Your function must include this interface (of type 'Interface.Microsoft_Excel_5.0_Object_Library.Range', for example) both as an input and output argument.

Notes:

- * On Windows systems with Excel software, **xlsread** reads any file format recognized by your version of Excel, including XLS, XLSX, XLSB, XLSM, and HTML-based formats.
- * If your system does not have Excel for Windows, **xlsread** operates in BASIC mode (see Input Arguments).
- * **xlsread** imports formatted dates as character vectors (such as '10/31/96'), except in BASIC mode. In BASIC mode, **xlsread** imports all dates as serial date

numbers. Serial date numbers in Excel use different reference dates than date numbers in MATLAB. For information on converting dates, see the documentation on importing spreadsheets.

```
*****  
*** xlsread is not recommended ***  
*****
```

Compatibility Considerations:

If the spreadsheet columns are mixed types, but has homogenous types along the columns, use READTABLE.

For reading numeric data, replace the following:

```
NUM = xlsread(FILE);
```

Instead Use:

```
NUM = readmatrix(FILE);
```

For reading text data, replace the following:

```
[~,TXT] = xlsread(FILE);
```

Instead Use:

```
TXT = readmatrix(FILE,"OutputType","char");
```

When reading RAW data, replace the following:

```
[~,~,RAW] = xlsread(FILE);
```

Instead Use:

```
RAW = readcell(FILE);
```

For specifying sheet and range, replace the following:

```
... = xlsread(FILE,SHEET,RANGE);
```

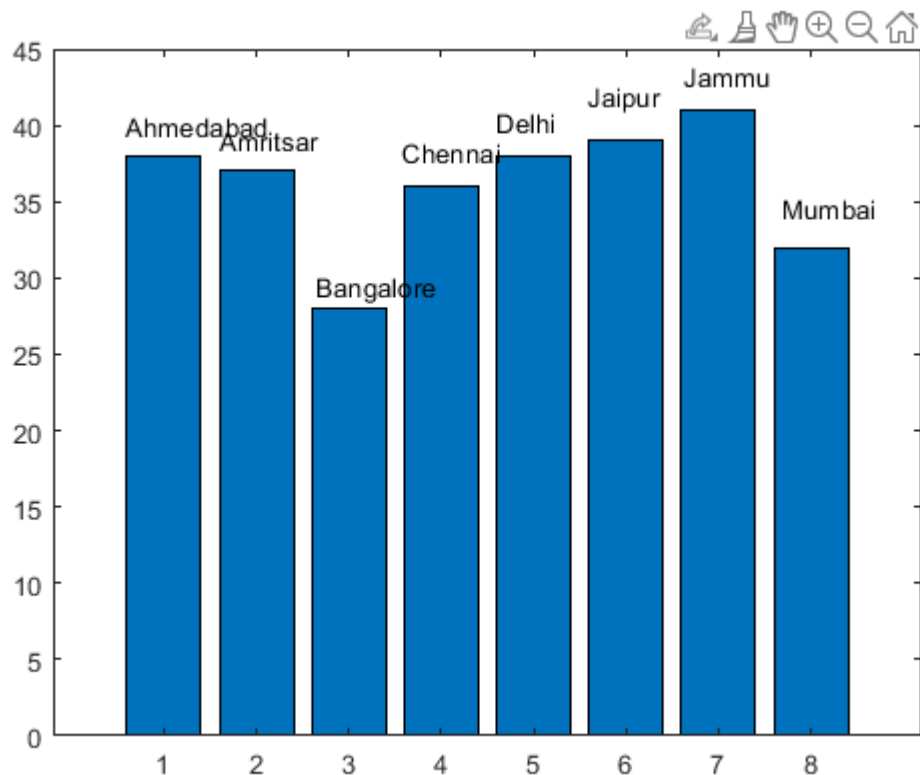
Instead use:

```
... = read<type>(FILE,"Sheet",SHEET,"RANGE",RANGE);
```

See also readtable, readmatrix, readcell, detectImportOptions

Documentation for xlsread

```
data = readtable("Lab_6_1.xlsx");  
bar(data.Max)  
  
cont =  
char('Ahmedabad','Amritsar','Bangalore','Chennai','Delhi','Jaipur','Jammu',  
     'Mumbai');  
  
for i = 1:8,  
    gtext(cont(i,:));  
end
```



Hence Jammu has recorded the Max Temperature of 41 Degrees

3.

a. Create an excel file with the following data:

| Roll No | Name | Accountancy | English | Maths | Economics | Business Studies |
|---------|----------|-------------|---------|-------|-----------|------------------|
| 1 | Akhilesh | 97 | 36 | 47 | 13 | 34 |
| 2 | Ruchi | 69 | 85 | 86 | 51 | 53 |
| 3 | Bhawna | 19 | 72 | 41 | 53 | 40 |
| 4 | Isha | 76 | 68 | 46 | 11 | 22 |
| 5 | Chetan | 55 | 31 | 56 | 99 | 93 |
| 6 | Neeti | 84 | 57 | 68 | 30 | 31 |
| 7 | Chanchal | 18 | 46 | 51 | 63 | 22 |
| 8 | Preeti | 93 | 93 | 31 | 93 | 20 |
| 9 | Richa | 33 | 89 | 55 | 46 | 69 |
| 10 | Manish | 21 | 27 | 84 | 82 | 96 |
| 11 | Karun | 13 | 48 | 27 | 26 | 38 |
| 12 | Madhur | 85 | 74 | 26 | 53 | 84 |
| 13 | Nitesh | 28 | 31 | 27 | 77 | 17 |

- Read the file in matlab using xlsread
- Create a function tot_marks to calculate total marks of each Student.
- Create function avg to calculate average marks of each student.
- Plot the marks obtained in step c. using

```
data_1 = readtable("Lab_6_2.xlsx")
```

Warning: Column headers from the file were modified to make them valid MATLAB identifiers before creating variable names for the table. The original column headers are saved in the VariableDescriptions property.

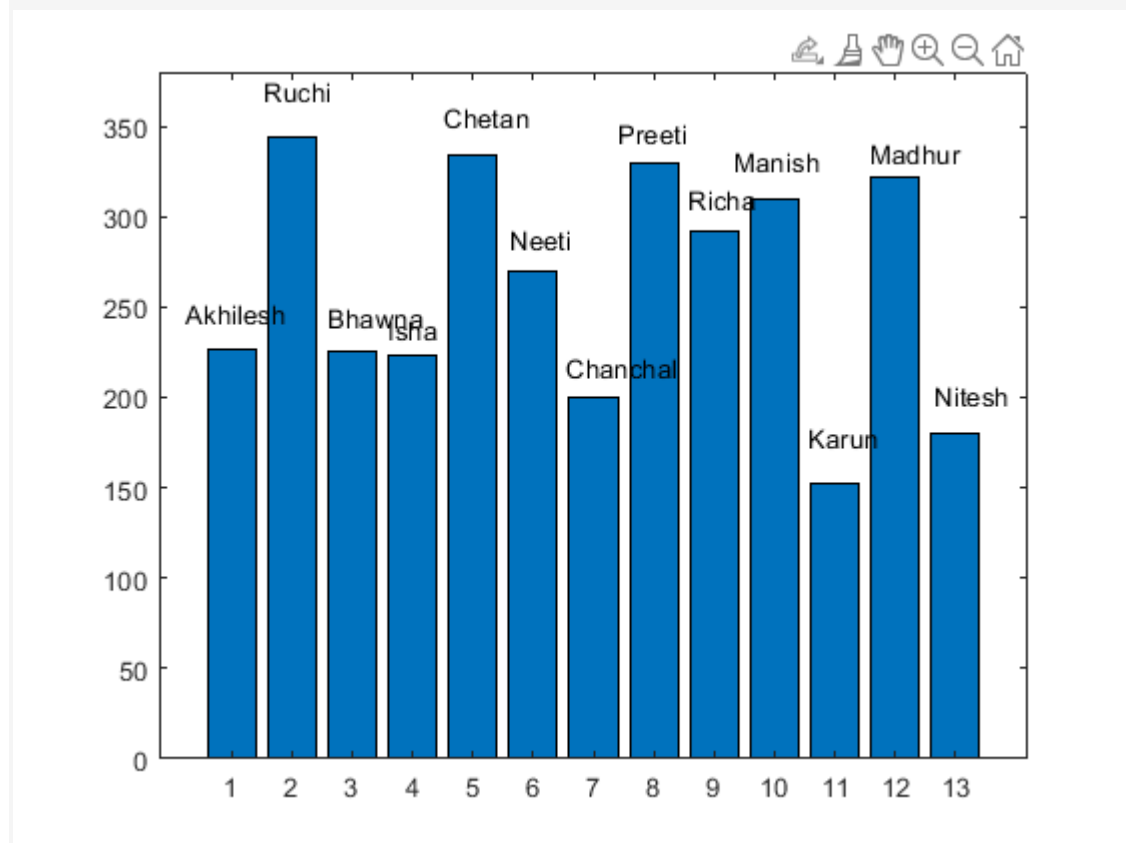
Set 'VariableNamingRule' to 'preserve' to use the original column headers as table variable names.

```
data_1 = 13x7 table
```

| | Roll No | Name | Accountancy | English | Maths | Economics | BusinessStudies |
|----|---------|------------|-------------|---------|-------|-----------|-----------------|
| 1 | 1 | 'Akhilesh' | 97 | 36 | 47 | 13 | 34 |
| 2 | 2 | 'Ruchi' | 69 | 85 | 86 | 51 | 53 |
| 3 | 3 | 'Bhawna' | 19 | 72 | 41 | 53 | 40 |
| 4 | 4 | 'Isha' | 76 | 68 | 46 | 11 | 22 |
| 5 | 5 | 'Chetan' | 55 | 31 | 56 | 99 | 93 |
| 6 | 6 | 'Neeti' | 84 | 57 | 68 | 30 | 31 |
| 7 | 7 | 'Chanchal' | 18 | 46 | 51 | 63 | 22 |
| 8 | 8 | 'Preeti' | 93 | 93 | 31 | 93 | 20 |
| 9 | 9 | 'Richa' | 33 | 89 | 55 | 46 | 69 |
| 10 | 10 | 'Manish' | 21 | 27 | 84 | 82 | 96 |
| 11 | 11 | 'Karun' | 13 | 48 | 27 | 26 | 38 |
| 12 | 12 | 'Madhur' | 85 | 74 | 26 | 53 | 84 |
| 13 | 13 | 'Nitesh' | 28 | 31 | 27 | 77 | 17 |

```
vars = ["Accountancy", "English", "Maths", "Economics", "BusinessStudies"];
data_1.TestMean = sum(data_1{:,vars},2);
bar(data_1.TestMean)
cont = char('Akhilesh', 'Ruchi', 'Bhawna', 'Isha', 'Chetan', ...
'Neeti', 'Chanchal', 'Preeti', 'Richa', 'Manish', 'Karun', 'Madhur', 'Nitesh');
ylim([0 380])
for i = 1:13,
    gtext (cont (i,:));
```

end



```
data_1.Average = mean(data_1[:,vars],2)
```

```
data_1 = 13x9 table
```

| | Roll No | Name | Accountancy | English | Maths | Economics | Business Studies | Test Mean | Average |
|---|---------|------------|-------------|---------|-------|-----------|------------------|-----------|---------|
| 1 | 1 | 'Akhilesh' | 97 | 36 | 47 | 13 | 34 | 227 | 45.4000 |
| 2 | 2 | 'Ruchi' | 69 | 85 | 86 | 51 | 53 | 344 | 68.8000 |
| 3 | 3 | 'Bhawna' | 19 | 72 | 41 | 53 | 40 | 225 | 45 |
| 4 | 4 | 'Ishfa' | 76 | 68 | 46 | 11 | 22 | 223 | 44.6000 |
| 5 | 5 | 'Chetan' | 55 | 31 | 56 | 99 | 93 | 334 | 66.8000 |
| 6 | 6 | 'Neeti' | 84 | 57 | 68 | 30 | 31 | 270 | 54 |
| 7 | 7 | 'Chanchal' | 18 | 46 | 51 | 63 | 22 | 200 | 40 |
| 8 | 8 | 'Preeti' | 93 | 93 | 31 | 93 | 20 | 330 | 66 |

| | | | | | | | | | |
|--------|----|--------------|----|----|----|----|----|-----|-------------|
| 9 | 9 | 'Rich a' | 33 | 89 | 55 | 46 | 69 | 292 | 58.4 000 |
| 1 0 | 10 | 'Mani sh' | 21 | 27 | 84 | 82 | 96 | 310 | 62 |
| 1 1 | 11 | 'Karu n' | 13 | 48 | 27 | 26 | 38 | 152 | 30.4 000 |
| 1 2 | 12 | 'Mad hur' | 85 | 74 | 26 | 53 | 84 | 322 | 64.4 000 |
| 1 3 | 13 | 'Nites h' | 28 | 31 | 27 | 77 | 17 | 180 | 36 |