LESSON: Introducing the sum function

FOCUS QUESTION: How can I transform the data to give more meaningful results?

This lesson shows you different ways to manipulate the data to give more meaningful analysis.

In this lesson you will:

- Call the MATLAB `sum` function to add up the rows or columns.
- Manipulate time series to plot on different time scales.
- Use the colon operator (`:`) to make an array into a single column.
- Combine arrays in different ways.
- Use the transpose operator (`'`) to flip an array.

Contents

- DATA FOR THIS LESSON
- SETUP FOR LESSON 3
- EXAMPLE 1: Load the NYC contagious disease data set (load .mat files)
- EXAMPLE 2: Calculate totals by year and by month (sum)
- EXAMPLE 3: Plot yearly total of measles by year (plot)
- EXAMPLE 4: Compare monthly totals of measles and mumps (multiple plots)
- EXAMPLE 5: Attempt to plot all the measles data as a single time series (linear representation)
- EXAMPLE 6: Correctly plot measles data as a single time series (transpose)
- EXAMPLE 7: Define the x-axis scale for a time series (subintervals a:inc:b)
- EXAMPLE 8: Plot measles cases as a time series, setting x-axis scale (computed scale)
- EXAMPLE 9: Plot a pie chart, comparing yearly measles counts for the first decade
- SUMMARY OF SYNTAX

DATA FOR THIS LESSON

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The data set contains the monthly totals of the number of new cases of measles, mumps, and chicken pox for New York City during the years 1931-1971. The file is organized into the following variables:</td>
</tr>
<tr>
<td></td>
<td>measles - an array containing the monthly cases of measles</td>
</tr>
</tbody>
</table>
NYCDiseases.mat

- mumps - an array containing the monthly cases of mumps
- chickenPox - an array containing the monthly cases of chicken pox
- years - a vector containing the years 1931 through 1971

The data was extracted from the Hipel-McLeod Time Series Datasets Collection, available at http://www.stats.uwo.ca/faculty/aim/epubs/mhsets/readme-mhsets.html.


SETUP FOR LESSON 3

- Make a new directory called SumFunction on your V: drive.
- Download the data file to your SumFunction directory.
- Create a SumFunctionLesson script file in your SumFunction directory.

EXAMPLE 1: Load the NYC contagious disease data set (load .mat files)

Create a new cell in which you type and execute:

```matlab
load NYCDiseases.mat; % Load the NYC disease data
```

You should see measles, mumps, chickenPox, and years variables in the Workspace Browser.

EXAMPLE 2: Calculate totals by year and by month (sum)

Create a new cell in which you type and execute:

```matlab
measlesByMonth = sum(measles, 1); % Sum along dimension 1 (column sums)
measlesByYear = sum(measles, 2); % Sum along dimension 2 (row sums)
measlesTotal = sum(measlesByMonth); % Overall total number of cases
```

You should see 3 variables in the Workspace Browser:

- measlesByMonth - row vector of 12 elements with total measles cases for each month
- measlesByYear - column vector of 41 elements with total measles cases for each year
- measlesTotal - a single value with overall total number of measles cases

EXERCISE 1: Find totals of other diseases

- Define a variable called mumpsByYear that contains the total case counts of mumps for each year.
- Define a variable called measlesAprilTotal that contains the total number of measles cases for the month of April.
- Define a variable called chickenPoxByYear that contains the total case counts for chicken pox in each year.
Define a variable called `chickenPoxTotal` that contains the overall total number of chicken pox cases.

**EXAMPLE 3: Plot yearly total of measles by year (plot)**

Create a new cell in which you type and execute:

```matlab
figure % Create a new figure window
plot(years, measlesByYear./1000) % Draw a line graph
xlabel('Year') % Label the x-axis
ylabel('Total cases (in thousands)') % Label the y-axis
title('NYC measles cases') % Put a title on the graph
```

You should see a Figure Window with a rescaled y-axis:

![Graph showing yearly total of measles cases in NYC](image)

**EXERCISE 2: Create a variable to hold counts of the 3 diseases by year.**
Define a new variable that holds a 41 x 3 array containing the total cases of the three diseases by year. Name the variable appropriately. The first column holds the measles total cases by year, the second column holds the mumps total cases by year, and the third column holds the chicken pox total cases by year.

**EXERCISE 3: Plot the variable defined in EXERCISE 2 in a new figure.**

**EXERCISE 4: Find total cases of the three diseases by year.**
Define a new variable that holds a 41 x 1 array containing the total number of cases of the three diseases by year.
EXAMPLE 4: Compare monthly totals of measles and mumps (multiple plots)

Create a new cell in which you type and execute:

```matlab
figure % Create a new figure window
hold on % Keep plots in same figure
plot(measlesByMonth./1000, '-rs') % Plot measles
plot(sum(mumps)./1000, '-ko') % Plot mumps
hold off
xlabel('Month') % Label the x-axis
ylabel('Total cases (in thousands)') % Label the y-axis
title('Measles and mumps in NYC: 1931-1971') % Title the graph
legend('Measles', 'Mumps') % Legend
```

You should see a Figure Window with a rescaled y-axis:

![Graph showing measles and mumps cases in NYC: 1931-1971 with a rescaled y-axis.]

EXERCISE 5: Add another plot to EXAMPLE 4 showing monthly totals of chicken pox.

EXAMPLE 5: Attempt to plot all the measles data as a single time series (linear representation)

Create a new cell in which you type and execute:
EXAMPLE 6: Correctly plot measles data as a single time series (transpose)

Create a new cell in which you type and execute:

```matlab
measlesFlip = measles'; % Flip measles to make rows into columns
figure % Create a new figure window
plot(measlesFlip(:)); % Draw a line graph
```

You should see a Figure Window with a single line graph:
EXAMPLE 7: Define the x-axis scale for a time series (subintervals a:inc:b)

Create a new cell in which you type and execute:

```
yearStart = 1931; % Start of the scale
yearInc = 1/12; % Scale has one month intervals
yearEnd = 1972 - yearInc; % End of the scale
yearScale = yearStart:yearInc:yearEnd; % Yearly scale (1 month incr)
```

You should see `yearStart`, `yearInc`, `yearEnd`, and `yearScale` variables in the Workspace Browser.

EXAMPLE 8: Plot measles cases as a time series, setting x-axis scale (computed scale)

Create a new cell in which you type and execute:

```
figure % Create a new figure window
plot(yearScale, measlesFlip(:,)/1000) % Draw a line graph
xlabel('Year') % Label the x-axis
ylabel('Cases (in thousands)') % Label the y-axis
title('NYC measles cases') % Put a title on the graph
```

You should see a Figure Window with a single line graph:
**EXERCISE 6**: Plot a graph of mumps as a single time series.

**EXAMPLE 9**: Plot a pie chart, comparing yearly measles counts for the first decade

Create a new cell in which you type and execute:

```matlab
yearLabel={'1931','1932','1933','1934','1935','1936','1937','1938','1939','1940'};
figure
pie(measlesByYear(1:10), yearLabel) % Draw a pie chart
title('Annual count of NYC measles cases (1931-1940)') % Title graph
```

You should see a Figure Window with a pie chart:
EXERCISE 7: Plot a pie chart of the second decade of measles by year.

SUMMARY OF SYNTAX

<table>
<thead>
<tr>
<th>MATLAB syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sum(A, 1)</code></td>
<td>Add the columns of the array A along dimension 1. If A is an n x m array, the result will be a 1 x m vector.</td>
</tr>
<tr>
<td><code>sum(A, 2)</code></td>
<td>Add the rows of the array A along dimension 2. If A is an n x m array, the result will be a n x 1 vector.</td>
</tr>
<tr>
<td><code>sum(A)</code></td>
<td>Add along the first non-singleton dimension of the array A.</td>
</tr>
<tr>
<td><code>[a, b, c]</code></td>
<td>Form an array with values of a, b and c placed side-by-side.</td>
</tr>
<tr>
<td><code>[a; b; c]</code></td>
<td>Form an array with values of a, b and c placed vertically end-to-end.</td>
</tr>
<tr>
<td><code>a:b</code></td>
<td>Form a row vector with the values a, a+1, ..., b.</td>
</tr>
<tr>
<td><code>a:inc:b</code></td>
<td>Form a row vector with the values a, a+inc, a+2*inc, ..., b. If b-a isn't evenly divisible by inc, the list stops before reaching b.</td>
</tr>
<tr>
<td><code>A(:)</code></td>
<td>Create a single column containing the columns of the array A positioned vertically end-to-end (i.e., the first column of A, followed by the second column of A, etc.).</td>
</tr>
<tr>
<td><code>A'</code></td>
<td>Create a new array in which the values of A are flipped along its main diagonal so that the rows become the columns and the columns become the rows. A' is called the transpose of A.</td>
</tr>
</tbody>
</table>