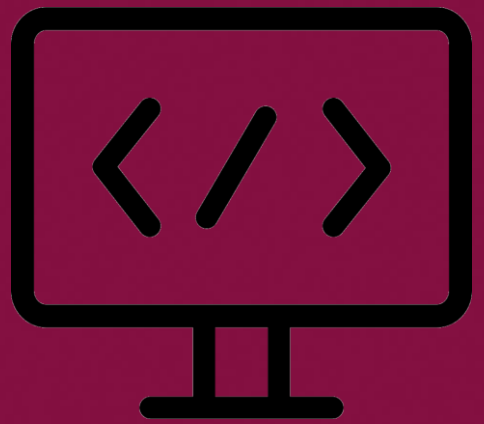
SEEE1022 INTRODUCTION TO SCIENTIFIC PROGRAMMING

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

CH10
Data Analysis



Dr. Mohd Saiful Azimi Mahmud (azimi@utm.my)
P19a-04-03-30, School of Electrical Engineering, UTM

OBJECTIVES

- After studying this chapter you should be able to:
  - Understand and use basic statistical MATLAB function.
  - Understand and use MATLAB sort function.

BASIC STATISTICAL ANALYSIS

BASIC STATISTICAL ANALYSIS

STATISTICAL ANALYSIS FUNCTIONS

- Below are some of the basic statistical analysis function available in MATLAB.

Function	Description
max	Maximum value
mean	Average or mean value
median	Median value
min	Smallest value
mode	Most frequent value
std	Standard deviation
var	Variance, which measures the spread or dispersion of the values

BASIC STATISTICAL ANALYSIS

COLUMN-ORIENTED DATASETS

- In MATLAB, datasets in a matrix are stored in column-oriented by default. Hence, in most MATLAB function, operation on column-oriented datasets does not require additional input.

EXAMPLE 1

```
>> x = [1 3; 2 5; 6 4]
x =
     1     3
     2     5
     6     4
>> mean(x)
ans =
     3     4

>> mean(x,2)
ans =
    2.0000
    3.5000
    5.0000
```

- Computing average values on column-oriented matrix does not require second input.
- Computing average values on row-oriented matrix require the second input to be set equals to 2

BASIC STATISTICAL ANALYSIS

OPERATION FOR THE ENTIRE MATRIX

EXAMPLE 2

To find statistical values for the entire matrix, use either colon operator or perform the function twice.

```
>> x = [1 3; 2 5; 6 4]
x =
     1     3
     2     5
     6     4
>> mean(x(:))
ans =
     3.5000

>> mean(mean(x))
ans =
     3.5000
```

- The colon operator converts matrix into column vector. Operation on vector does not require the second input.
- Performing the function twice will first compute column-oriented data that return a vector. Then the operation is done to the vector that return a scalar.

BASIC STATISTICAL ANALYSIS

TEMPERATURE AT 3 CITIES

EXAMPLE 3

- Load a temperature data for 3 cities from the file Temp1.mat and send the data to a user-defined `datastat()` function shown on the next slide for basic statistical analysis of the data.
- Temp1.mat consists of two variables; a matrix `temp` where every column represents each of the three cities temperature values in ℃ and a scalar `daypersample` showing number of days between each of the temperature reading.
- Function `datastat()` has 5 inputs and return 5 statistical data in a single matrix. The last input to the function is a string that determine how the data will be displayed.
- Since `temp` is already set as column-oriented for each city, all of the statistical functions used in the `datastat()` does not need the second input.
- While `temp` is the y-axis data, the x-axis data can be created according to the `daypersample` value. The x-axis data can be created as below:

```
xdata = 1:daypersample:length(temp(:,1))
```

BASIC STATISTICAL ANALYSIS

TEMPERATURE AT 3 CITIES

EXAMPLE 3

Below is the MATLAB code for the function `datastat()`

```
function [statisticVal] = datastat(xdata,ydata,xname,yname,dispOption)
    % Input...
    % Output...
    N = size(ydata,2);
    statisticVal = zeros(5,N);
    statisticVal(1,:) = mean(ydata);
    statisticVal(2,:) = var(ydata);
    statisticVal(3,:) = median(ydata);
    statisticVal(4,:) = max(ydata);
    statisticVal(5,:) = min(ydata);
    switch dispOption
        case 'NONE'
        case 'CW'
            dispdata(statisticVal,yname)
        case 'PLOT'
            dispdata(statisticVal,yname)
            plotdata(xdata,ydata,xname,yname)
        otherwise
            disp('DISPLAY OPTION IS NOT VALID')
    end
end
```

All of the statistical function does not have second input. Thus, make sure to set `ydata` as a column-oriented when using function `datastat()`

BASIC STATISTICAL ANALYSIS

TEMPERATURE AT 3 CITIES

EXAMPLE 3

The code continues here where there are two local functions…

```
function dispdata(stat,yname)
    N = size(stat,2);
    overall = [mean(stat(1,:)) mean(stat(2,:))...
               median(stat(3,:)) max(stat(4,:)) min(stat(5,:))];
    disp([yname ' data:-'])
    disp('              Average  Variance    Median       Max
Min')
    fprintf('Dataset %d :%10.2f%10.2f%10.2f%10.2f%10.2f\n',...
            [1:N;stat])
    fprintf('Overall   :'),fprintf('%10.2f',overall),fprintf('\n')
end

function plotdata(x,y,XLABEL,YLABEL)
    plot(x,y,'-o','LineWidth',1,'MarkerSize',4)
    xlabel(XLABEL), ylabel(YLABEL)
end
```

All of the statistical function within the function `dispdata()` does not need second input or set the input data as column-oriented since they are dealing with vectors.

BASIC STATISTICAL ANALYSIS

TEMPERATURE AT 3 CITIES

EXAMPLE 3

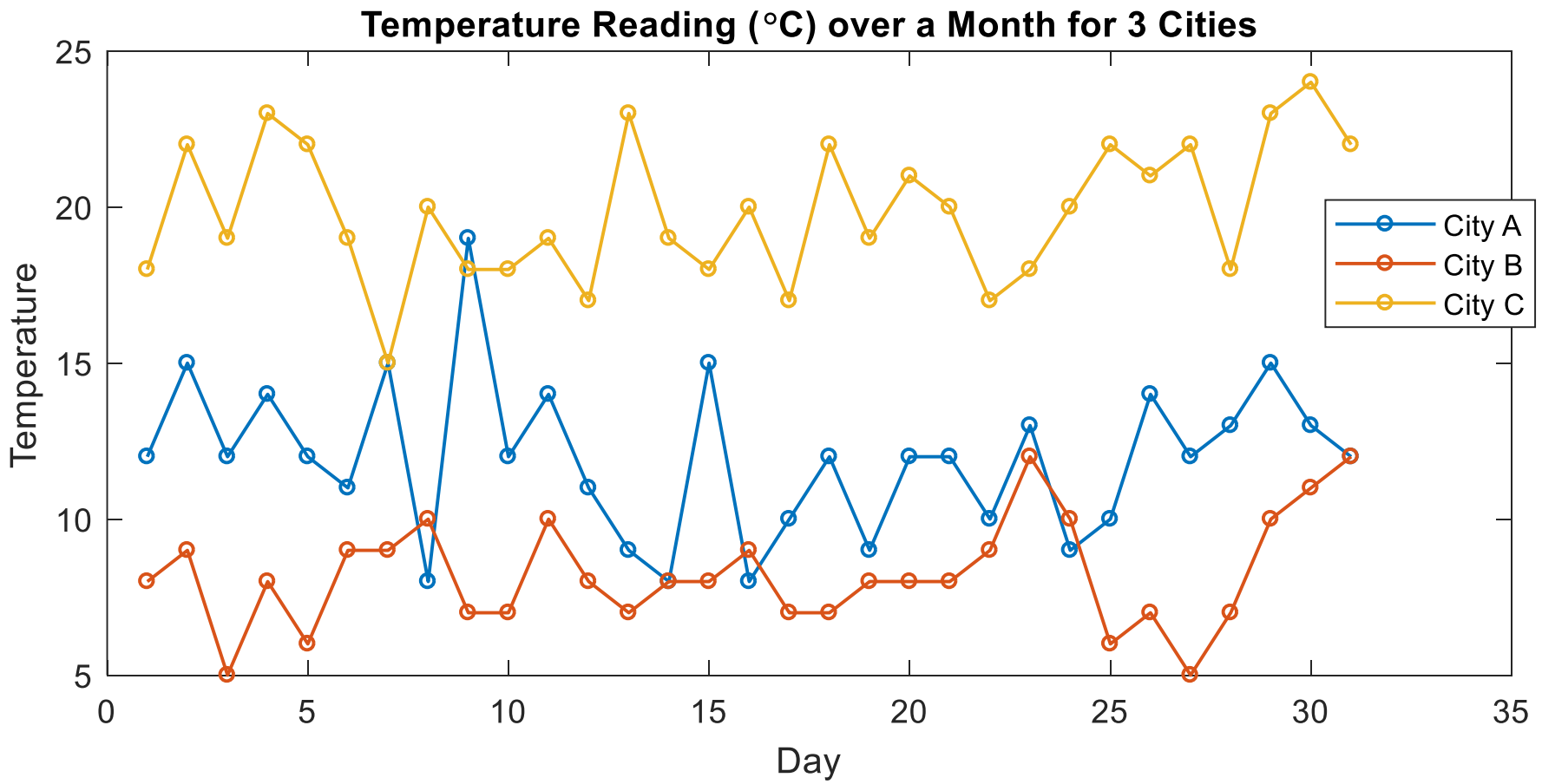
Below is how the data is loaded and passed to the `datastat()` function that returns the statistical values of the data.

```
>> load dataTemp1
>> xdata = 1:daypersample:length(temp(:,1));
>> y = datastat(1:31,temp,'Day','Temperature','NONE');
>>
>> y = datastat(1:31,temp,'Day','Temperature','PLOT');
Temperature data:-
              Average    Median       Max       Min
Dataset 1 :     11.97     12.00     19.00      8.00
Dataset 2 :      8.23      8.00     12.00      5.00
Dataset 3 :     19.87     20.00     24.00     15.00
Overall   :     13.35     12.00     24.00      5.00
>>
>> legend('City A','City B','City C')
>> title('Temperature Reading over a Month for 3 Cities')
```

BASIC STATISTICAL ANALYSIS

TEMPERATURE AT 3 CITIES

EXAMPLE 3



DATA SORTING

DATA SORTING

SORT

Syntax

```
[B,I] = sort(A,dim,direction)
```

Description

A	:	Data to be sorted.
dim	:	Dimension to operate along. 1 for column-oriented (default) and 2 for row-oriented.
direction	:	Sorting direction. ‘ascend’ (default) or ‘descend’.
B	:	Sorted data.
I	:	Sort index.

DATA SORTING

DATA-ORIENTATION

EXAMPLE 4

```
>> x = [6 3 7 2 8];
>> y = sort(x)
y =
     2     3     6     7     8

>> X = [3 5 2; 6 2 7]
X =
     3     5     2
     6     2     7

>> Ycol = sort(X)
Ycol =
     3     2     2
     6     5     7

>> Yrow = sort(X,2)
Yrow =
     2     3     5
     2     6     7
```

- Since MATLAB stored data as column-oriented by default, the second input, `dim` for the `sort()` function is only needed when sorting a data to row-oriented.
- Similar to other functions, the dim input is not needed for vector type of data.

DATA SORTING

SORT DIRECTION

EXAMPLE 5

```
>> X = [3 5 2; 6 2 7]
X =
     3     5     2
     6     2     7

>> Ycolasc = sort(X)
Ycolasc =
     3     2     2
     6     5     7

>> Ycoldes = sort(X,'descend')
Ycoldes =
     6     5     7
     3     2     2

>> Yrowdes = sort(X,2,'descend')
Yrowdes =
     5     3     2
     7     6     2
```

- '`ascend`' is the default sort direction for `sort()` function. Thus no need to specify the direction input when sorting data ascendingly.

DATA SORTING

SORT INDEX

EXAMPLE 6

Given below are temperature values for 9 consecutive days at a city. Find the 2 highest and 2 lowest temperature values. Also find which day these highest and lowest values occur.

$$x = [19 \quad 20 \quad 18 \quad 17 \quad 20 \quad 23 \quad 25 \quad 24 \quad 23]$$

Solution

- Find the 2 highest and 2 lowest temperature values can be easily done using `sort()` function twice; first with descending order and then ascending order.
- Day, although not explicitly define by any vector from the problem, it is implicitly known as the index of the vector x. Thus, the second output argument for `sort()` function can be used to find the respective 2 highest and 2 lowest temperature.

DATA SORTING

SORT INDEX

EXAMPLE 6

Below is the MATLAB code for Example 6

```
x = [19 20 18 17 20 23 25 24 23];
[xsortasc,iasc] = sort(x);
[xsortdes,ides] = sort(x,'descend');
N = 2;
Xlow = xsortasc(1:N);
Daylow = iasc(1:N);
Xhigh = xsortdes(1:N);
Dayhigh = ides(1:N);

fprintf('Highest %d\nno.   Temp    Day\n',N)
fprintf('%3d %6.2f %6d\n',[1:N;Xhigh;Dayhigh])
fprintf('Lowest %d\nno.   Temp    Day\n',N)
fprintf('%3d %8.2f %6d\n',[1:N;Xlow;Daylow])
```

```
Highest 2
no.    Temp     Day
  1      25       7
  2      24       8
Lowest 2
no.    Temp     Day
  1      17       4
  2      18       3
```

DATA SORTING

OUTLIERS

EXAMPLE 7

- Outliers are values in a set of data that are considered to have significant dissimilarity compared to the rest of the data. One approach to detect the dissimilarity is by checking statistical values of the dataset.
- Below is an example of an algorithm to remove N outliers from a dataset:

Start

Input: $data$, N

1) Compute difference between $data$ and its average, $d = |data - data_{avg}|$
2) Sort desceding the d and called it d_{sort}. Also find the corresponding sort index and called it i_{sort}.
3) Get the first N elements of i_{sort} and called it i_N.
4) Remove outlier of $data$ according to index i_N.

Output: $data$

End

- Write a MATLAB function for the above pseudocode.

DATA SORTING

OUTLIERS

EXAMPLE 7

Below is the MATLAB code for Example 7

```
function data = remoutliers(data,N)
d = abs(data - mean(data));
[dsort,isort] = sort(d,'descend');
iN = isort(1:N);
data(iN) = [];
```

```
>> x = [2 3 6 1 0 3 4 1 1 4 9 3 1 2 2];

>> xcleaned = remoutliers(x,1)
xcleaned =
        2    3    6    1    0    3    4    1    1    4    3    1    2    2

>> xcleaned = remoutliers(x,2)
xcleaned =
        2    3    1    0    3    4    1    1    4    3    1    2    2
```

DATA SORTING

DATA ANALYSIS DRILL

1) Given $x = [2,3,2]$ and $y = [4,1,3]$, does running MATLAB code `mean([x;y])` returns similar output compared to `mean([mean(x) mean(y)])`? Explain your answer.

2) If the answer in Question 1 is not similar, how to modify the vector concatenation of x and y of the first MATLAB code so that it will return the same value with the second MATLAB code.

3) Given vector x, Does `std(x(:))` and `std(std(x))` always return similar output? Explain your answer.

4) From Example 3, describe in detail the inputs and output of the `datastat()` function. Refer MATLAB documentation on any MATLAB function as an example on how the input and output are describe.

5) From Example 3, list down code lines in local function `dispdata()` when the code line "`statisticVal=zeros(5,N);`" within the function `datastat()` is change to "`statisticVal=zeros(N,5);`". Then, made necessary modification to the listed codes.

6) Write comments for each code line for Example 3, 6 and 7.

7) Rewrite MATLAB script in Example 6 when the temperature readings are taken once every two days.

8) Write a MATLAB script for the following pseudo-code. Use outliers removal function from Example 7 for step 2:

Start
Input: $data, T$
1) Compute variance, v of $data$
2) If v is greater than T, remove 10% outliers from $data$.
3) Repeat step 1 until $v \leq T$.

Output: $data$
End

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

univteknologimalaysia utm_my utmofficial

Thank You

www.utm.my

innovative • entrepreneurial • global