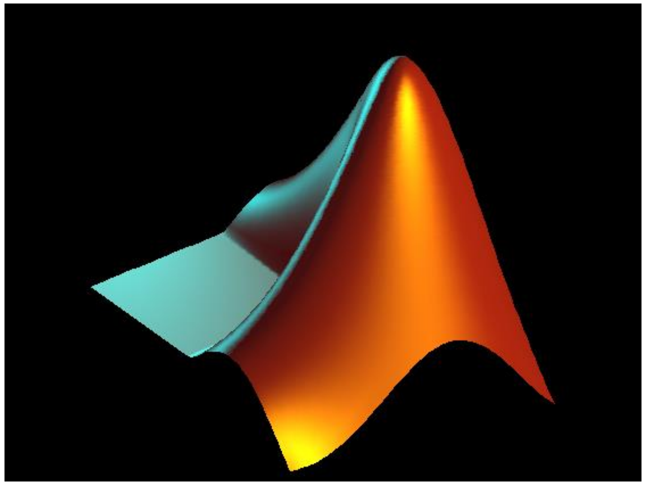
Data Handling



Topic 8

outline

- ASCII and binary data
- saving MATLAB data in MATLAB format: .mat files
- using the save and load functions
- importing numeric data from spreadsheets: cut/paste
- importing spreadsheet data in table format
- storing data in dataset format
- importing image files
- 8-bit colour images
- basic image processing examples

binary data and ASCII (text) data

Binary data consist of a series of 1s and 0s. One binary digit is called a bit. The most common binary (digital) data storage unit a sequence of 8 bits called byte.

Executable programs, image, audio and video files and numeric data are normally stored in binary format. Programs stored as binary files run very quickly – the downside is that they are not human-readable so they cannot be corrected or modified in their original format.

ASCII (American Standard Code for Information Interchange) is a code for representing English characters as numbers. Most computers store plain (i.e. unformatted) text in ASCII format.

An ASCII file is a text file in which each byte represents one character. The standard ASCII character set uses only 7 bits for each character, which allows for a total of 128 unique characters (see table here). Notice that the first digit is always 0.

ASCII to binary conversion

Hello and good morning!

in binary format is

010010000110010101101100011011000110111
1001000000011000010110111001100100001000
0001100111011011110110111101100100000100
000011011010110111101110010011011100110
1001011011100110011100100001

Each group of 8 (with a leading 0) represents 1 character (as shown →).

Can you find the *space* character?

01001000
01100101
01101100
01101100
01101111
00100000
01100001
01101110
01100100
00100000
01100111
01101111
01101111
01100100
00100000
01101101
01101111
01110010
01101110
01101001
01101110
01100111
00100001

mat files

mat-files store Workspace variables in *binary format*.

```
>> save myfile
```

writes all arrays in the Workspace into a file named **`myfile`**. The extension is .mat by default. The usual naming rules apply to mat-files as well.

To save selected variables (say **`x1`** and **`x2`**), include the names of the variables after the filename

```
>> save filename x1 x2
```

Do not use commas here. MATLAB will interpret the following

```
>> save myfile x1, x2
```

as two separate statements. It will save **`x1`** into myfile and then displays variable **`x2`** in the Command Window.

importing data

MATLAB includes functions to import

- mat files (matfile)
- text (ASCII) data (textread)
- spreadsheets (e.g. xlsread)
- scientific data (e.g. cdfread – common data format)
- images (e.g. imread)
- audio and video (e.g. audioread)

(and some less common file formats, e.g. XML)

importing numeric data– copy/paste

- Numeric data can be copied and pasted into the Workspace editor.
- Blank cells will be padded with zeros.
- This process will work for any spreadsheet editor, e.g. Excel, OpenOffice, Google Spreadsheets

Download the Sheffield weather data form MOLE (*sheffieldweather.xlsx*) and copy the Max Temperature dataset into the Workspace editor.

Do not include column labels, only numeric data.

Save the data in your current directory.

Create the following plots:
July_mean(Year)
Summer_mean(Year)

the Import Tool

- Click on the **Import Data** button in the Home menu.
- Select *sheffieldweather.xlsx* – the file will open in the Import Tool.
- Now we can select the range of the data to import and the format of MATLAB output.
- Select the option to import each column as a different variable. Variables will be named after the original column labels.
- Use CTRL-Click to select the columns to import. You can grab the borders and drag them up and down to restrict or expand your selection.
- Notice the NaN values in the 2015 rows (padding for blanks)
- Explore the data visualisation tools in the Plot tab. To create a quick line or scatter plot of two or more arrays (of the same size), click on Variable 1 (x-axis) first, then CTRL-Click on Variable 2 (and so on) and then select a plot tool.

the table format

- Importing data in **table format** allows MATLAB to summarise all data (columns) at once
- Select *sheffieldweather.xlsx* – and choose the TABLE output
- **table** is a data type in MATLAB – for more information click here
- Try the following

```
>> format compact
>> summary(sheffieldweather)
```

- The summary function calculates the minimum, median, and maximum score for each numerical variable
- The command format compact displays the output without blank lines. To switch back to the default display format, use format loose

sample datasets

The Statistics and Machine Learning Toolbox comes with a collection of sample data in dataset format.

The dataset type is described here:
http://uk.mathworks.com/help/stats/dataset-arrays.html

The list of sample datasets can be found here:
http://uk.mathworks.com/help/stats/_bq9uxn4.html

To load the hospital dataset into the Workspace, simply type

```
>> load hospital
```

Double-click on hospital in the Workspace to view the dataset.

Try the summary function and examine the different summaries created for numerical, categorical and logical variables. For a detailed analysis of this dataset, go to http://uk.mathworks.com/help/stats/dataset-array-columns.html

importing image files

The following example uses Penguins.jpg, one of the sample images included in the Sample Pictures folder in Windows 7. You can use any image from your Current Directory or any other image on MATLAB's Search Path.

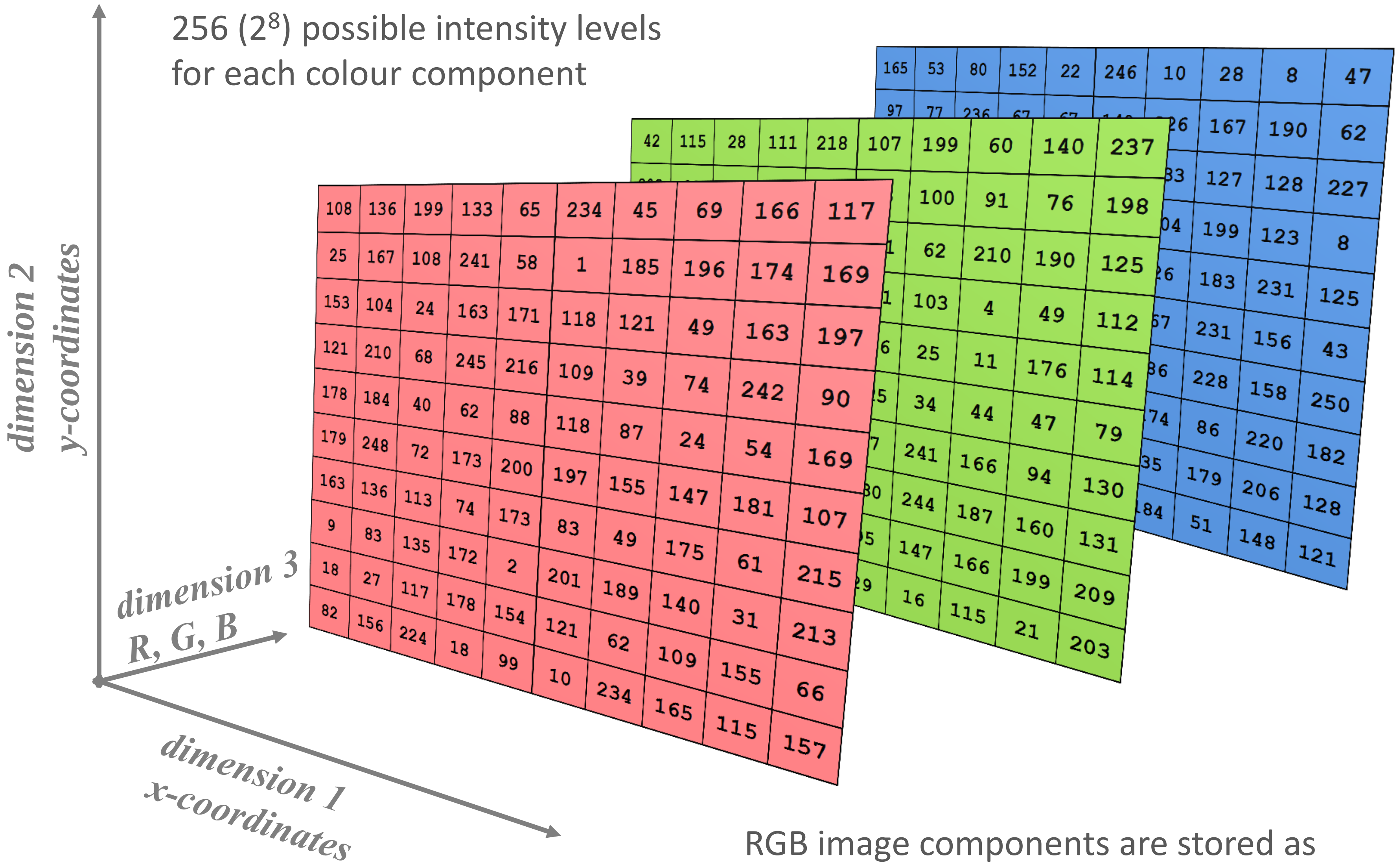
Use the imread function to import the image.

```
>> imp=imread('Penguins.jpg');
>> imshow(imp) % display the image, imagesc(imp) also works

>> size(imp)
ans =
     768   1024   3  % imp is a 3D array: x, y, color (R,G or B,)

>> iR=imp(:,:,1); % select R component only
% iR (matrix) gives the R level for each pixel

>> iR_range=[min(iR(:)),max(iR(:))];
>> disp(iR_range)
    0   255
```

R, G and B values are now specified by integers between 0 and 256 (uint8 format) instead of type double values from the interval [0,1].

8-bit images

256 (2^8) possible intensity levels for each colour component



108	136	199	133	65	234	45	69	166	117
25	167	108	241	58	1	185	196	174	169
153	104	24	163	171	118	121	49	163	197
121	210	68	245	216	109	39	74	242	90
178	184	40	62	88	118	87	24	54	169
179	248	72	173	200	197	155	147	181	107
163	136	113	74	173	83	49	175	61	215
9	83	135	172	2	201	189	140	31	213
18	27	117	178	154	121	62	109	155	66
82	156	224	18	99	10	234	165	115	157

RGB image components are stored as integers from [0, 255]

basic image manipulation example

```
% script to show R, G and B components of image separately

% import image data
f=input('filename(string): '); % 'Penguin.jpg'
img = imread(f);

showR=img; showG=img; showB=img; % create 3 copies of the image

showR(:,:,2:3)=0; % remove G and B
showG(:,:,[1,3])=0; % remove R and B
showB(:,:,1:2)=0; % remove R and G

figure % opens 1st figure window
imshow(img)
pause(1), figure % opens separate figure window after 1 sec
imshow(showR)
pause(1), figure
imshow(showG)
pause(1), figure
imshow(showB)
```

original image; Red, Green and Blue components