

# Programming Supplement

Machine Learning – CISC 5800  
Dr Daniel Leeds

## Matrix indexing

- Start counting at 1  
 $\text{matrix1} = [4 \ 8 \ 12; 6 \ 3 \ 0; -2 \ -7 \ -12];$   
 $\text{npMat} = \text{np.array}(\text{Mat})$   
 $\text{npMat}[1,2]$
- Last row/column can also be designated by keyword “end”  
 $\text{matrix1}(1,\text{end}) \rightarrow 12$   
 $\text{npMat}[0,-1]$
- Colon indicates counting up by increment
  - $[2:10] \rightarrow [2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10]$   
 $\text{range}(2,11)$
  - $[3:4:19] \rightarrow [3 \ 7 \ 11 \ 15 \ 19]$   
 $\text{range}(3,19,4)$ $\text{matrix1}(2,1:3) \rightarrow [6 \ 3 \ 0]$   
 $\text{npMat}[1,0:3]$

3

## Programming in Matlab/Python: Data types

- Numbers: -8.5, 0, 94
- Characters: 'j', '#', 'K'      'j', "k", '#', "3"
- Groups of numbers/characters – placed in between []  
  - [5 10 12; 3 -4 12; -6 0 0] – spaces/commas separate columns,  
semi-colons separate rows
  - \text{Mat}=[[5, 10, 12], [3, -4, 12], [-6, 0, 0]]  
 $\text{npMat} = \text{np.array}(\text{Mat})$
  - \text{import numpy as np}

2

## Vector/matrix functions

- ```
vec1=[9, 3, 5, 7]; matrix2=[4.5 -3.2; 2.2 0; -4.4 -3];
• mean    mean(vec1) -> 6          \text{vec1}=\text{np.array}([9, 3, 5, 7])
• min     min(vec1) -> 3          \text{matrix2}=\text{np.array}([[4.5, -3.2], [2.2, 0], [-4.4, -3]])
• max     max(vec1) -> ?          • \text{np.mean} (\text{vec1})
• std     std(vec1) -> 2.58        • \text{np.min}(\text{vec1}) -> 3
• length  length(vec1) -> ?       • \text{np.max} (\text{vec1}) -> ?
• size    size(matrix2) -> [3 2];  • \text{np.std}(\text{vec1}) -> 2.58
   • \text{len}(\text{vec1}) -> ?
   • (\text{matrix2}).\text{shape} -> [3 2];
```

4

## Extra syntax notes

- Semicolons suppress output of computations:

```
> a=4+5
a =
9
> b=6+7;
>
% starts a comment for the line (like // in C++)
.* , ./, .^ performs element-wise arithmetic
>c=[2 3 4]./[2 1 2]
>c =
[1 3 1]
>
```

5

## Data: .mat files

- save** filename variableNames

- load** filename

- Confirm correct directories:

- pwd – show directory (print working directory)
- cd – change directory
- ls – list files in directory

```
import scipy.io
varIn = scipy.io.loadmat('file.mat')
X=varIn['matVarX']
```

7

## Variables

- who, whos – list variables in environment
- Comparisons:
  - Like C++: ==, <, >, <=, >=
  - Not like C++: not ~, and &, or |
- Conditions:
  - if(...), end;
- Loops:
  - while(...), end;
  - for x=a:b, end;

6

## Define new functions: .m files

- Begin file with function header:

```
function output = function_name(input)
```

```
def func_name(input):
    statement1
    statement2
    ...
    return output
```

- Can allow multiple inputs/outputs

```
function [output1, output2] = function_name(input1, input2, input3)
```

8