Guess Two-Thirds of the Average Game Quiz
Comment on Vectorization
Admin
A Boolean variable, also called *logical* variable type in MATLAB, is a variable with two possible values *true* and *false*.

A Boolean variable is stored as either 1 for *true* or 0 for *false*. 
**Indicators**

**Vectorization**

**Functions**

---

**Indicator Functions**

**Math**

- Indicator functions, also called dummy variables, are functions that return 1 if a condition is satisfied and 0 if the condition is not satisfied.

1. $x == y$ is the indicator of $x = y$, meaning \[ \begin{align*} 
1 & \quad \text{if } x = y \\
0 & \quad \text{if } x \neq y 
\end{align*} \]

2. $x \sim= y$ is the indicator of $x \neq y$, meaning \[ \begin{align*} 
1 & \quad \text{if } x \neq y \\
0 & \quad \text{if } x = y 
\end{align*} \]

   $x \\neq y$ does not work in MATLAB.

3. $x > y$, $x >= y$ are indicators of $x > y$ and $x \geq y$.

4. $x < y$, $x <= y$ are indicators of $x < y$ and $x \leq y$. 
Other Logical Functions

Code

- $\sim$ is not: $\sim 0$ is 1 and $\sim 1$ is 0.

- $\&$ is and: $0 \& 0$ is 0, $0 \& 1$ is 0, $1 \& 0$ is 0, $1 \& 1$ is 1.

- $|$ is or: $0 \mid 0$ is 0, $0 \mid 1$ is 1, $1 \mid 0$ is 1, $1 \mid 1$ is 1.

- $==, \sim=, <, <=, >, >=, \sim, \&,$ can be applied element-wise to a vector directly.
Short Circuit Evaluation

Code

- `&&` is and, but only works on scalars.
- `||` is or, but only works on scalars.
- `&&` and `||` use short-circuit evaluation, for example, when evaluating `a && b`, if `a` is false, then `b` will not be evaluated, and when evaluating `a || b`, if `a` is true, then `b` will not be evaluated.
**Vector Reduction Logical Functions**

**Code**

- `any(x)` returns whether any of the elements in the matrix or vector `x` is non-zero.
- `all(x)` returns whether all of the elements in the matrix or vector `x` is non-zero.
- `find(x)` finds the index of all the non-zero elements in the vector `x`.
- `find(x, 1)` finds the index of the first non-zero element in the vector `x`. 
Other Reduction Functions

Code

- $\text{sum}(x)$ and $\text{prod}(x)$ compute the sum and product of the elements in a matrix or vector $x$.
- $\text{sum}(x, 1)$ and $\text{prod}(x, 1)$ compute the column sums and products of the elements in a matrix $x$, for example, $\text{sum}([1\ 2;\ 3\ 4],\ 1)$ returns the column sums $[4\ 6]$.
- $\text{sum}(x, 2)$ and $\text{prod}(x, 2)$ compute the row sums and products of the elements in a matrix $x$, for example, $\text{sum}([1\ 2;\ 3\ 4],\ 2)$ returns the row sums $[3\ 7]$.
- $\text{mean}(x)$ computes the average of the numbers in a matrix or vector $x$.
- $\text{max}(x)$ and $\text{min}(x)$ compute the maximum and minimum of the elements in a matrix or vector $x$. 
Indicator, Quiz Grade

Quiz
Indicator, Grade Point Average

Quiz
Indicator, Letter Grade
Quiz
Indicator, Letter Grades
Quiz
A function \( y = f(x) \) is a mapping from a list of inputs \( x \), also called arguments or parameters, to a list of outputs \( y \).

The previous lectures covered many built-in functions in MATLAB, for example, \( \log \) has 1 input and 1 output, \( + \) has 2 inputs and 1 output, and \( \text{size} \) has 1 input and 2 outputs.

New functions can be defined in .m files and used in commands.
A function with name $f$ should be put in a file named $f.m$.

The first line of the file is `function y = f(x)` or `function [y1, y2, ...] = f(x1, x2, ...)`, where $y$ is the name or names of the variables to return, and $x$ is the list of arguments of the function.

The second line of the file is usually comments describing what the function does. Comments start with `%` the line after `%` is not executed by the program.

The last line of the file should be `end`, but it can be omitted.
Multiple functions can be defined in the same file $f.m$, but only $f$ can be used outside the file in commands.

The functions in $f.m$ that is not $f$ are helper functions.
Function Example, Addition

Code

- The addition function $x + y$ is usually written in infix notation (argument 1, then function name, then argument 2).
- The following function is the addition function in prefix notation (function name, then argument 1, then argument 2).

```plaintext
function z = add(x, y)
    z = x + y;
end
```

- $add(1, 2)$ returns 3.
The linear combination of $x$ and $y$ with coefficients $u$ and $v$ is $ux + vy$.

Sometimes, $u, v$ are not specified, so the default value $u = v = 1$ is used.

```matlab
function z = lincom(x, y, u, v)
    arguments
    x; y; u = 1; v = 1;
    end
    z = u * x + v * y;
    end
```

(arguments block is also used for input validation. More detail in a later lecture.)
Function Example, Linear Combination Too

Code

1 \textit{function} \ z = \textit{lincom}(x, y, u, v) \\
2 \textit{arguments} \\
3 \hspace{1em} x; y; u = 1; v = 1; \\
4 \textit{end} \\
5 \hspace{1em} z = u \times x + v \times y; \\
6 \textit{end} \\

- \textit{lincom}(1, 2, 3, 4) \textit{returns} 11 . \\
- \textit{lincom}(1, 2, 3) \textit{returns} 5 . \\
- \textit{lincom}(1, 2) \textit{returns} 3 .
Multiple values can be returned from a function, 
\[ y_1, y_2, \ldots, y_n \] = f(x) \text{ stores the value of } i\text{-th output in } y_i \text{ for } i = 1, 2, \ldots, n \text{ and } f(x) \text{ only returns first output.}

```matlab
function [mx, mn] = mnxn(x)
    mx = max(x);
    mn = min(x);
end
```

- \( \text{mnxn([1, 2, 3]) returns 3} \)
- \( [a \ b] = \text{mnxn([1, 2, 3]) sets } a \text{ to 3 and } b \text{ to 1} \).
<table>
<thead>
<tr>
<th>Indicators</th>
<th>Vectorization</th>
<th>Functions</th>
</tr>
</thead>
</table>

**Functions, Vector Output**

Quiz
Functions, Multiple Outputs

Quiz
Blank Slide