MFE Programming Workshop Class 2

Rob Richmond

December 3, 2015

UCLA Anderson

Questions

Any questions before we start?

Optimizing code

- In general Matlab is faster when you use vectorized functions/operators instead of loops
- Many functions/operators are vectorized (they operate over individual elements)
- Lets look at last weeks lab

Plotting

Lets go over some plotting facilities

Anonymous functions

- Functions in Matlab can be created inline for simple tasks using @

Example

```
myfunc = @(x) exp(x^2);
myval = myfunc(1.1);
ans = myval
```

Optimization (1)

- Matlab has an entire toolbox dedicated to optimization
- Some possible uses are
  - Root finding: `fzero`, `fsolve`
  - Unconstrained optimization: `fminunc`
  - Constrained optimization: `fmincom`
  - Quadratic programing: `quadprog`

Optimization (2)

- Lets look at an example of root finding
- Suppose we want x s.t. $5 - e^{x^2} = x^2$
- We convert this to the root finding problem $5 - e^{x^2} - x^2 = 0$

Example

```
myfunc = @(x) 5-exp(x^2)-x^2;
myval = fsolve(myfunc,2)
ans = myval
```

Statistical functions

- Basic statistics functions are labeled as you would expect: `mean`, `var`, etc.
- `cov` returns the covariance matrix of a matrix consisting of column vecotrs of observations
- removing `nan` values can be useful: `nanmean`, `nancov`