TECHNISCHE UNIVERSITÄT DRESDEN

Introduction to Matlab

Debugging, fitting and some statistics

Dario Cuevas and Vahid Rahmati

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Statistics

Built-in basic statistical functions in matlab:

function	what it does…
mean	Gives the mean of the input
var	Gives the variance
std	Gives the standard deviation
cov	Gives the covariance matrix
corr	Gives the correlation matrix
median	Gives the median

Built-in random number generators in matlab:

function	what it does…
rand	samples from 0 to 1 with equal probability
randn	samples from the normal distribution

Some nice-to-know functions

isempty: tells you whether an array is empty
find: returns the indices of the values that meet the conditions
sort: sorts the values in a vector
permute: changes the order of the dimensions of an array
randperm: returns the numbers in a random order
reshape: reshapes a matrix (see help)
isnan/isinf: checks if the input is or includes NaN/Inf
try/catch:
break: leaves the current for/while/if. Can also stop a script
return: same as break, but only works in functions and scripts
pause: stops the execution for a specified time
repmat: repeats the input a specified number of times
bsxfun:
ismember: checks if an array contains a given number
round, ceil, floor: rounding tools
sprintf, fprintf: print to strings and the screen, respectively
input: takes an input from the keyboard
max/min: finds the maximum/minimum value of an array.

Fprintf and sprintf

fprintf('The value of x is %f', x) will print on the command window 'The value of x is __' and in the blank space it will print whatever the value of x is.
You can put many of them:
fprintf('x = %f, y = %f, z = %f', x,y,z), and it will print them in that order.
The type of variable has to be indicated:

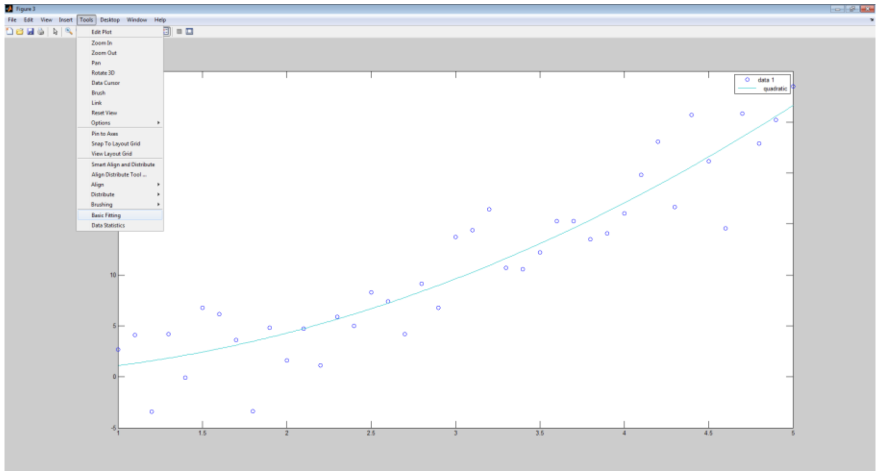
%f	Double
%d	Integer
%s	String

sprintf works in the same way, except that the output is a string, so:
mystring = sprintf(''The value of x is %f', x) will create a string mystring whose value is 'The value of x is__', where the blanks are filled with the value of variable x.

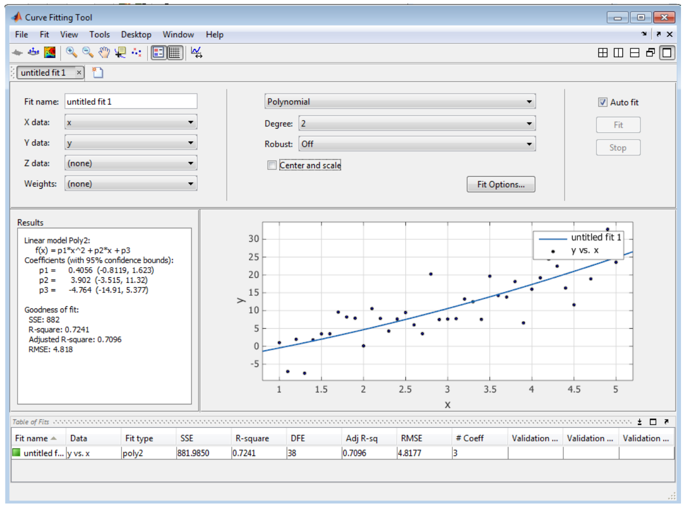
You can specify format for the numbers presented. For example, 'x = %2.0f' will write the value of the variable with 2 characters in total, 0 of which are after the decimal point. '%05.2f' will write two numbers after the decimal, 5 characters in total and it will fill with zeroes on the left if there are no more numbers available.

Fitting

Matlab can do basic fitting of points with a function using the Figure GUI.



cftool offers more andvanced posibilities for fitting



Fit name: untitled fit 1
X data: x
Y data: y
Z data: (none)
Weights: (none)

Polynomial
Degree: 2
Robust: Off
[ ] Center and scale
[✓] Auto fit

```
Linear model Poly2:
    f(x) = p1*x^2 + p2*x + p3
Coefficients (with 95% confidence bounds):
      p1 =     0.4056  (-0.8119, 1.623)
      p2 =      3.902  (-3.515, 11.32)
      p3 =     -4.764  (-14.91, 5.377)

Goodness of fit:
  SSE: 882
  R-square: 0.7241
  Adjusted R-square: 0.7096
  RMSE: 4.818
```

Fit name	Data	Fit type	SSE	R-square	DFE	Adj R-sq	RMSE	# Coeff	Validation ...	Validation ...	Validation ...
🟩 untitled f...	y vs. x	poly2	881.9850	0.7241	38	0.7096	4.8177	3			

Debugging

Types of errors: typos, syntax, logic

There are two possibilities in matlab’s debugging system:

1. breakpoints
2. stop when error/warning

Useful tools while in debug mode:
next step
step in
step out
continue

Exercise:
Find the functions in the website. Run the main.m. The output should be a number.
Find and correct the error using the debug mode.

Miscellaneous exercises

1. Create a random number generator that outputs integers from 1 to 100. (rand, floor/ceil)
2. Generate a random vector with 1,000 entries. Create a piece of code that selects 30 random entries form this vector, without repetition. (randperm)
3. Ask the user for some input, then print to the screen 'The user said: %s', where %s represents whatever input the user gave. (input, fprintf)
4. randn returns random numbers with the normal distribution of mean 0 and variance 1. Create a code that returns numbers of mean 1 instead (var = 1).
5. Printing a progress indicator. Create a loop for ii=1:100. Inside this loop, put a command pause(0.1) and a progress indicator, that is, code that tells you how much of the loop has elapsed. The output of this code (with fprintf) should be 'Progress: XX%', where XX is the percentage so far. The code should not print each percentage in a different line, but overwrite the previous one. Use ' \b' to delete the last character to be printed. ' \b\b' would erase the last two characters.