Loading and Visualizing Data ...

Program paused. Press enter to continue.

Cost at theta = [1 ; 1]: 303.993192

(this value should be about 303.993192)

Program paused. Press enter to continue.

Gradient at theta = [1 ; 1]: [-15.303016; 598.250744]

(this value should be about [-15.303016; 598.250744])

Program paused. Press enter to continue.

Iteration 2 | Cost: 2.237391e+01

Program paused. Press enter to continue.

Iteration 3 | Cost: 0.000000e+00

Iteration 3 | Cost: 9.860761e-32

Iteration 8 | Cost: 3.286595e+00

Iteration 13 | Cost: 2.842678e+00

Iteration 25 | Cost: 1.315405e+01

Iteration 36 | Cost: 1.944396e+01

Iteration 15 | Cost: 2.009852e+01

Iteration 18 | Cost: 1.817286e+01

Iteration 9 | Cost: 2.260941e+01

Iteration 16 | Cost: 2.326146e+01

Iteration 20 | Cost: 2.431725e+01

Iteration 2 | Cost: 2.237391e+01

Iteration 2 | Cost: 0.000000e+00

Iteration 5 | Cost: 9.860761e-31

Iteration 18 | Cost: 1.412195e-01

Iteration 9 | Cost: 2.625825e+01

Iteration 11 | Cost: 2.217027e+01

Iteration 23 | Cost: 1.862410e+01

Iteration 27 | Cost: 2.372155e+01

Iteration 8 | Cost: 2.091082e+01

Iteration 36 | Cost: 1.872262e+01

Iteration 9 | Cost: 1.717055e+01

Iteration 20 | Cost: 1.566566e+01

Iteration 8 | Cost: 1.491869e+01

# Training Examples Train Error Cross Validation Error

1 181.188753 253.098982

2 92.598337 139.727476

3 34.474749 49.773898

4 37.713149 30.855158

5 32.017334 31.887001

6 27.412310 32.668299

7 24.677941 29.377233

8 24.420951 29.297406

9 23.215440 28.953669

10 22.670744 28.956963

11 22.389953 28.824951

12 22.373906 29.050121

Program paused. Press enter to continue.

Normalized Training Example 1:

1.000000

-0.362141

-0.755087

0.182226

-0.706190

0.306618

-0.590878

0.344516

-0.508481

Program paused. Press enter to continue.

Iteration 200 | Cost: 1.466650e-01

Iteration 2 | Cost: 0.000000e+00

Iteration 10 | Cost: 0.000000e+00

Iteration 25 | Cost: 6.573841e-32

Iteration 24 | Cost: 3.055357e-28

Iteration 200 | Cost: 1.879561e-10

Iteration 200 | Cost: 4.273063e-05

Iteration 200 | Cost: 2.128252e-02

Iteration 200 | Cost: 7.101701e-02

Iteration 200 | Cost: 1.784403e-01

Iteration 200 | Cost: 1.421305e-01

Iteration 200 | Cost: 1.285036e-01

Iteration 200 | Cost: 1.466650e-01

Iteration 2 | Cost: 0.000000e+00

Iteration 11 | Cost: 0.000000e+00

Iteration 115 | Cost: 5.390550e-30

Iteration 200 | Cost: 7.449075e-26

Iteration 200 | Cost: 1.341339e-09

Iteration 200 | Cost: 8.220965e-05

Iteration 200 | Cost: 2.997389e-08

Iteration 200 | Cost: 1.868459e-02

Iteration 200 | Cost: 1.101232e+00

Iteration 200 | Cost: 1.020790e+00

Iteration 200 | Cost: 7.798276e-01

Iteration 200 | Cost: 1.283298e+00

Polynomial Regression (lambda = 0.000000)

# Training Examples Train Error Cross Validation Error

1 133.837122 162.401555

2 133.057791 166.615525

3 40.508510 75.613910

4 43.152144 14.333897

5 2.207599 47.459744

6 5.010173 46.335075

7 1.258243 50.398175

8 0.747305 71.926715

9 0.335098 33.685894

10 0.178227 36.790032

11 0.167130 143.597435

12 0.146665 19.672637

Program paused. Press enter to continue.

warning: imaginary part of complex colon arguments is ignored

warning: called from

validationCurve at line 35 column 19

ex5 at line 204 column 38

warning: imaginary part of complex colon arguments is ignored

warning: called from

validationCurve at line 35 column 19

ex5 at line 204 column 38

error: operator \*: nonconformant arguments (op1 is 10x1, op2 is 9x1)

error: called from

linearRegCostFunction at line 28 column 6

trainLinearReg>@<anonymous> at line 13 column 21

fmincg at line 77 column 11

trainLinearReg at line 19 column 7

validationCurve at line 35 column 19

ex5 at line 204 column 38

>> [5\*5; 5\*5]

ans =

25

25

>> [er,ers]=ans

er =

25

25

error: element number 2 undefined in return list

>> [free freee]

error: 'free' undefined near line 1, column 2

>> A = 4;5]

error: parse error:

syntax error

>>> A = 4;5]

^

>> A = [4;5]

A =

4

5

>> B = [5;4]

B =

5

4

>> free = [A B]

free =

4 5

5 4

>> free = [A, B]

free =

4 5

5 4

>> free = A, B

free =

4

5

B =

5

4

>> [er1,er2]=A,B

er1 =

4

5

error: element number 2 undefined in return list

>> free = A, B

free =

4

5

B =

5

4

>> free[1]

error: parse error:

syntax error

>>> free[1]

^

>> free(1)

ans = 4

>> free(2)

ans = 5

>> free = A B

error: parse error:

syntax error

>>> free = A B

^

>> free = A; B

B =

5

4

>> free = (A B)

error: parse error:

syntax error

>>> free = (A B)

^

>> free(1)

ans = 4

>> free(2)

ans = 5

>> free(1,2)

error: free(\_,2): out of bound 1 (dimensions are 2x1)

>> [one,two]=A , B

one =

4

5

error: element number 2 undefined in return list

>> [one two]=A , B

one =

4

5

error: element number 2 undefined in return list

>> [one two]=A B

error: parse error:

syntax error

>>> [one two]=A B

^

>> [one two]=A ; B

error: element number 2 undefined in return list

>> [one two]=A , B

one =

4

5

error: element number 2 undefined in return list

>> one two = ~A , B

error: variable "one" used as function in command style expression

>> one two = A , B

error: variable "one" used as function in command style expression

>> one= A , B

one =

4

5

B =

5

4

>> [one two] =A

one =

4

5

error: element number 2 undefined in return list

>> [one] =(A,B)

error: parse error:

syntax error

>>> [one] =(A,B)

^

>> [one] =(A B)

error: parse error:

syntax error

>>> [one] =(A B)

^

>> one =(A B)

error: parse error:

syntax error

>>> one =(A B)

^

>> one =(A, B)

error: parse error:

syntax error

>>> one =(A, B)

^

>> one =(A; B)

error: parse error:

syntax error

>>> one =(A; B)

^

>> theta2=[1;0.5;2]

theta2 =

1.0000

0.5000

2.0000

>> theta2(2)

ans = 0.5000

>> theta2[2]

error: parse error:

syntax error

>>> theta2[2]

^

>> cd ..

>> ls

octave1 octave2 octave3 octave4 octave5

>> cd octave1

>> ls

computeCost.m ex1\_multi.m gradientDescent.m normalEqn.m warmUpExercise.m

computeCostMulti.m ex1data1.txt gradientDescent2.m plotData.m

ex1.m ex1data2.txt gradientDescentMulti.m submit.m

ex1.pdf featureNormalize.m lib token.mat

>> load('ex1data1'

)

error: load: unable to find file ex1data1

>> load('ex1data1')

error: load: unable to find file ex1data1

>> ls

computeCost.m ex1\_multi.m gradientDescent.m normalEqn.m warmUpExercise.m

computeCostMulti.m ex1data1.txt gradientDescent2.m plotData.m

ex1.m ex1data2.txt gradientDescentMulti.m submit.m

ex1.pdf featureNormalize.m lib token.mat

>> load('ex1data1.txt')

>> load('ex1data2.txt')

>> whos

Variables visible from the current scope:

variables in scope: top scope

Attr Name Size Bytes Class

==== ==== ==== ===== =====

A 2x1 16 double

B 2x1 16 double

J 1x1 8 double

X 12x1 96 double

X\_poly 12x9 864 double

X\_poly\_test 21x9 1512 double

X\_poly\_val 21x9 1512 double

Xtest 21x1 168 double

Xval 21x1 168 double

ans 1x1 8 double

er 2x1 16 double

er1 2x1 16 double

error\_train 12x1 96 double

error\_val 12x1 96 double

ex1data1 97x2 1552 double

ex1data2 47x3 1128 double

free 2x1 16 double

grad 2x1 16 double

i 1x1 8 double

lambda 1x1 8 double

m 1x1 8 double

mu 1x8 64 double

one 2x1 16 double

p 1x1 8 double

sigma 1x8 64 double

theta 9x1 72 double

theta2 3x1 24 double

y 12x1 96 double

ytest 21x1 168 double

yval 21x1 168 double

Total is 1001 elements using 8008 bytes

>> X

X =

-15.9368

-29.1530

36.1895

37.4922

-48.0588

-8.9415

15.3078

-34.7063

1.3892

-44.3838

7.0135

22.7627

>> y

y =

2.1343

1.1733

34.3591

36.8380

2.8090

2.1211

14.7103

2.6142

3.7402

3.7317

7.6277

22.7524

>> thetanormal = pinv(X'\*y)\*X'\*y;

>> gradientDescent(X,y,thetanormal,1.6,9)

theta = -964.20

theta = 1.3091e+06

theta = -1.7767e+09

theta = 2.4114e+12

theta = -3.2729e+15

theta = 4.4422e+18

theta = -6.0291e+21

theta = 8.1830e+24

theta = -1.1106e+28

ans = -1.1106e+28

>> three=[one,two]=gradientDescent(X,y,thetanormal,1.6,9);

theta = -964.20

theta = 1.3091e+06

theta = -1.7767e+09

theta = 2.4114e+12

theta = -3.2729e+15

theta = 4.4422e+18

theta = -6.0291e+21

theta = 8.1830e+24

theta = -1.1106e+28

>> one

one = -1.1106e+28

>> two

two =

3.9485e+08

7.2735e+14

1.3399e+21

2.4682e+27

4.5467e+33

8.3757e+39

1.5429e+46

2.8422e+52

5.2357e+58

>> three[1]

error: parse error:

syntax error

>>> three[1]

^

>> three(1)

ans = -1.1106e+28

>> three(2)

error: three(2): out of bound 1 (dimensions are 1x1)

>> three(0)

error: three(0): subscripts must be either integers 1 to (2^63)-1 or logicals

>> [one,two]=gradientDescent(X,y,thetanormal,1.6,9);

theta = -964.20

theta = 1.3091e+06

theta = -1.7767e+09

theta = 2.4114e+12

theta = -3.2729e+15

theta = 4.4422e+18

theta = -6.0291e+21

theta = 8.1830e+24

theta = -1.1106e+28

>> one

one = -1.1106e+28

>> two

two =

3.9485e+08

7.2735e+14

1.3399e+21

2.4682e+27

4.5467e+33

8.3757e+39

1.5429e+46

2.8422e+52

5.2357e+58

>>

