```
1 # Solving Bevington example 6.1 with Octave
2
3 printf( "Bevington Example 6.1\n" )
4 strftime ("%Y-%m-%d %H:%M:%S", localtime (time ()))
5
6 # design matrix
7 printf( "design matrix:\n" )
8 \parallel A = [11; 12; 13; 14; 15; 16; 17; 18; 19]
10 # Define data
11 printf( "data vector:\n" )
T = [15.6; 17.5; 36.6; 43.8; 58.2; 61.6; 64.2; 70.4; 98.8]
13
14 # solve least squares problem
printf( "least squares solution\n" )
16 | printf( "xls = A \\ T:\n" )
17 \| xls = A \setminus T
18
19 # residual error vector
20 residual = A * xls - T;
21 | printf( "residual error vector = A * xls - T n")
22
23 t2 = dot( residual, residual );
  printf( "least total squared error t2 = residual . residual = %d\n", t2 )
24
25
26 # compute Gram matrix
27 \parallel W = transpose(A) * A;
28
29 # invert Gram matrix
30 \parallel \text{Winv} = \text{inv}(\ \text{W}\ );
31 values = diag( Winv );
32
33 # measure design matrix
34 \parallel m = rows, n = columns
| [m, n] = size (A);
36 g = sprintf('%d', [m, n]);
  fprintf('matrix dimensions: %s\n', g);
38
  printf( "\ncompute error elements:\n" )
  printf( "sigma = sqrt( t2 / ( m - n ) * values ):\n" )
40
  sigma = sqrt(t2 / (m - n) * values)
41
42
43 printf( "\n# # # Compare Octave values to exact values\n" )
  printf( "\nFit parameters\n" )
45
  printf( "\nerror in intercept and slope values\n" )
  printf( "numericError = xls - [ 1733 / 360; 1129 / 120 ]\n" )
47
48 numericError = xls - [ 1733 / 360; 1129 / 120 ]
```

```
49
50 printf( "\nerror in intercept and slope values in machine epsilon\n" )
51||printf( "epsError = numericError ./ eps( 1.0 )\n" )
  epsError = numericError ./ eps( 1.0 )
53
  printf( "\nError parameters\n" )
54
55 | printf( "\nintercept and slope sigmas\n" )
56 printf( "numericError = sigma - sgrt( [ 108297055; 3419907 ] / 35 ) /
  360\n")
  numericError = sigma - sqrt( [ 108297055; 3419907 ] / 35 ) / 360
57
58
  printf( "\nerror in intercept and slope sigmas in machine epsilon\n" )
59
  printf( "epsError = numericError ./ eps( 1.0 )\n" )
  epsError = numericError ./ eps( 1.0 )
61
62
  ## dantopa@Quaxolotl.local:least-squares $ pwd
63
  ## /Volumes/T7-Touch/repos/github/jop/octave/genesis/least-squares
65 ## dantopa@Quaxolotl.local:least-squares $ octave-cli wtf.m
66 ## Bevington Example 6.1
67 ## ans = 2022-09-06 20:41:57
68 ## design matrix:
  ## A =
69
70
         1
  ##
             1
71
         1
             2
72 ##
         1
             3
73 ##
         1
             4
  ##
74
         1
             5
75 | ##
76 ##
         1
             6
             7
  ##
         1
77
         1
             8
  ##
78
         1
             9
  ##
79
80
  ## data vector:
81
82 | ## T =
83
         15.600
84 ##
85 ##
         17.500
         36,600
86 ##
87 ##
         43.800
  ##
         58.200
88
89 ##
         61.600
         64.200
90 ##
        70.400
91 ##
         98.800
92
  ##
93
  ## least squares solution
  |## xls = A \setminus T:
```

```
## xls =
96
97
          4.8139
98
   ##
   ##
          9.4083
99
100
   ## residual error vector = A * xls - T
101
   ## least total squared error t2 = residual . residual = 316.658
102
   ## matrix dimensions: 9 2
103
104
   ## compute error elements:
105
   ## sigma = sqrt( t2 / ( m - n ) * values ):
106
   ## sigma =
107
108
   ##
          4.8862
109
          0.8683
   ##
110
111
112
                Compare Octave values to exact values
113
114
   ## Fit parameters
115
116
   ## error in intercept and slope values
117
   |## numericError = xls - [ 1733 / 360; 1129 / 120 ]
   ## numericError =
119
120
         -1.0658e-14
121 ##
   ##
         1.7764e-15
122
123
124
  ## error in intercept and slope values in machine epsilon
125
   ## epsError = numericError ./ eps( 1.0 )
126
   ## epsError =
127
128
   ##
         -48
129
   ##
           8
130
131
132
   ## Error parameters
133
134
   ## intercept and slope sigmas
135
   ## numericError = sigma - sgrt( [ 108297055; 3419907 ] / 35 ) / 360
136
   ## numericError =
137
138
          8.8818e-16
139
   ##
          1.1102e-16
   ##
140
141
142
   ## error in intercept and slope sigmas in machine epsilon
143
```

/Volumes/repos/github/jop/pdf/bevington/least-squares.m Page 4/4 Saved: 12/5/22, 21:28:03 Printed for: Daniel M. Topa

```
## epsError = numericError ./ eps( 1.0 )
## epsError =

146
147 ## 4.0000
148 ## 0.5000
```