

Demo document with computer code

HPL

Apr 8, 2016

1 Data file

Suppose we have some data in a file. The final result of including this file with `@@@CODE mydat.txt` (which implies a code environment starting with `!bc dat`) looks like this:

```
1  #      A      B      C      D      E
2  -0.5253  -0.9315  -0.3427  -0.1613  -0.8472
3  -0.9740  -0.2558  -0.5622  -0.7635  -0.0914
4   0.9216   0.7702  -0.4818   0.2155   0.2967
```

2 Complete program and terminal output

The following program (which breaks a page) reads the data in the file and performs analysis (typeset with `!bc pypro`):

```
1  import numpy as np
2
3  def readfile(filename):
4      """Read tabular data from file and return as numpy array."""
5      f = open(filename, 'r')
6      data = [] # list of rows in table
7      for line in f:
8          if line.startswith('#'):
9              continue # drop comment lines
10             numbers = [float(w) for w in line.split()]
11             data.append(numbers)
12     return np.array(data)
13
```

```

14 def analyze(data):
15     """Return statistical measures of an array data."""
16     return np.mean(data), \
17         np.std(data), \
18         np.corrcoef(data)
19
20 if __name__ == '__main__':
21     data = readfile('mydat.txt')
22     # Treat each column as a variable
23     m, s, c = analyze(data.transpose())
24     print """
25     mean=%f
26     st.dev=%f
27     correlation matrix:
28     %s
29     """ % (m, s, c)

```

The output becomes (typeset with !bc sys):

```

1 Terminal> python fileread.py
2
3 mean=-0.006005
4 st.dev=0.583542
5 correlation matrix:
6 [[ 1.          0.0509676  0.52406366  0.20964645  0.1574504 ]
7  [ 0.0509676   1.          -0.30920845 -0.12129049  0.7611538 ]
8  [ 0.52406366 -0.30920845  1.          0.49355806 -0.42263817]
9  [ 0.20964645 -0.12129049  0.49355806  1.          -0.38286589]
10 [ 0.1574504   0.7611538  -0.42263817 -0.38286589  1.          ]]

```

3 Code snippet

Fortran 77 is also sometimes handy. Snippets in that language are typeset inside !bc fcod environments.

Fortran code box.

$$r_i = ca_i, \quad i = 1, \dots, n$$

```
1      subroutine process(a, n, c, r)
2  C    This subroutine returns array r = c*a
3      integer n
4      real*8 a(n), c, r(n)
5      integer i
6      do i = 1,n
7          r(i) = c*a(i)
8      end do
9      return
0      end
```