

Demo document with computer code

HPL

Feb 1, 2015

1 Data file

Suppose we have some data in a file:

#	A	B	C	D	E
	-0.5253	-0.9315	-0.3427	-0.1613	-0.8472
	-0.9740	-0.2558	-0.5622	-0.7635	-0.0914
	0.9216	0.7702	-0.4818	0.2155	0.2967
	0.6217	0.6100	-0.3846	-0.7904	0.9166
	0.1006	-0.3162	0.3841	0.5241	-0.6530
	0.6207	-0.9299	0.4837	0.5755	-0.6024
	0.4278	-0.0014	0.8184	0.9382	-0.1449
	-0.9178	0.2612	-0.7532	0.3901	-0.0075
	0.2134	0.6217	0.0545	0.6980	-0.2172
	-0.9529	0.8989	-0.1969	-0.3079	0.0389
	0.8311	0.0145	0.4215	-0.5451	-0.3415

2 Program

The following program (which breaks a page) reads the data in the file and performs analysis:

```
1  #!/usr/bin/env python
   import numpy as np
   4
   def readfile(filename):
       """Read tabular data from file and return as numpy array."""
       7  f = open(filename, 'r')
       data = [] # list of rows in table
       for line in f:
           10  if line.startswith('#'):
               continue # drop comment lines
               numbers = [float(w) for w in line.split()]
```

```

13     data.append(numbers)
    return np.array(data)

16 def analyze(data):
    """Return statistical measures of an array data."""
    return np.mean(data), \
19         np.std(data), \
        np.corrcoef(data)

22 if __name__ == '__main__':
    data = readfile('mydat.txt')
    # Treat each column as a variable
25    m, s, c = analyze(data.transpose())
    print ""
    mean=%f
28    st.dev=%f
    correlation matrix:
    %s
31    """ % (m, s, c)

```

The output becomes

Terminal

```

Terminal> python fileread.py

mean=-0.006005
st.dev=0.583542
correlation matrix:
[[ 1.          0.0509676  0.52406366  0.20964645  0.1574504 ]
 [ 0.0509676   1.          -0.30920845 -0.12129049  0.7611538 ]
 [ 0.52406366 -0.30920845  1.          0.49355806 -0.42263817]
 [ 0.20964645 -0.12129049  0.49355806  1.          -0.38286589]
 [ 0.1574504   0.7611538  -0.42263817 -0.38286589  1.          ]]

```

3 Fortran example

Here is an example of a Fortran 77 snippet:

```

2  C Return array r = c*a
    subroutine process(a, n, c, r)
    integer n
    real*8 a(n), c, r(n)
5   integer i
    do i = 1,n
        r(i) = c*a(i)
    enddo
end

```

8

```
end do  
return  
end
```