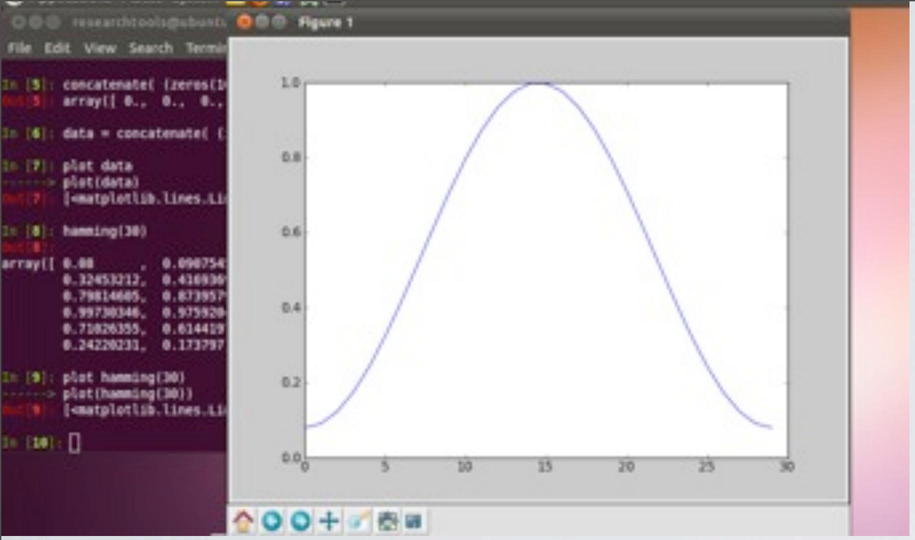
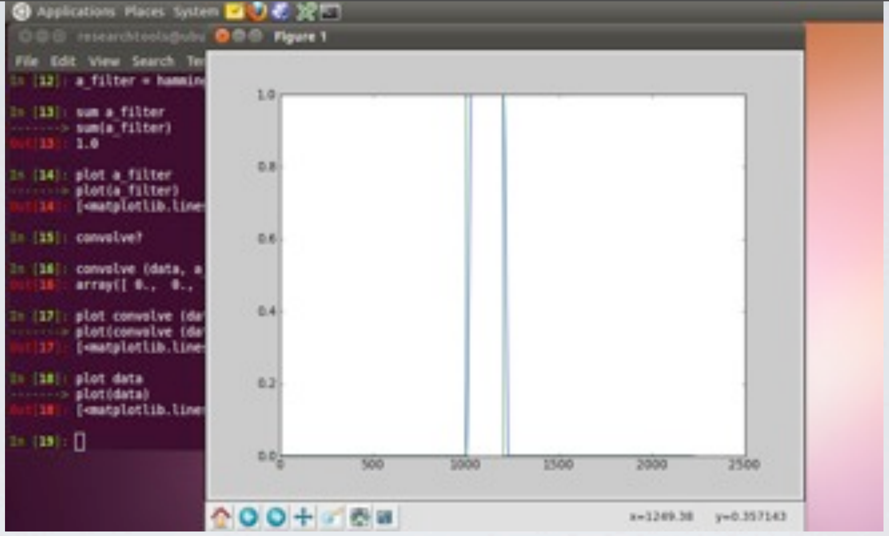
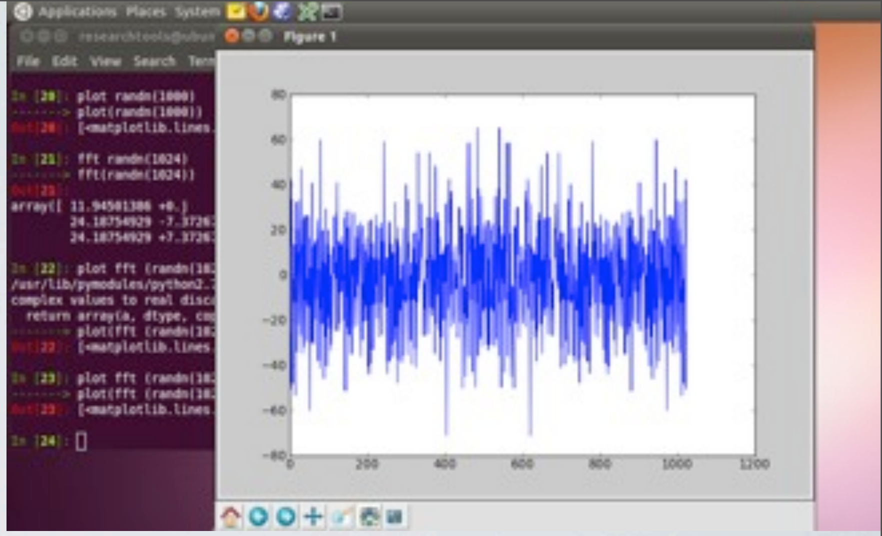


```
In [5]: concatenate( (zeros(1
Out[5]: array([ 0.,  0.,  0.,

In [6]: data = concatenate( (

In [7]: plot data
------> plot(data)
Out[7]: [<matplotlib.lines.Li

In [8]: hamming(30)
Out[8]:
array([ 0.08      ,  0.090754
        0.32453212,  0.416930
        0.79814605,  0.873957
        0.99730346,  0.975920
        0.71026355,  0.614419
        0.24220231,  0.173797

In [9]: plot hamming(30)
------> plot(hamming(30))
Out[9]: [<matplotlib.lines.Li

In [10]: 
```



```
In [12]: a_filter = hammin

In [13]: sum a_filter
-------> sum(a_filter)
Out[13]: 1.0

In [14]: plot a_filter
-------> plot(a_filter)
Out[14]: [<matplotlib.line

In [15]: convolve?

In [16]: convolve (data, a
Out[16]: array([ 0.,  0.,

In [17]: plot convolve (da
-------> plot(convolve (da
Out[17]: [<matplotlib.line

In [18]: plot data
-------> plot(data)
Out[18]: [<matplotlib.line

In [19]: 
```



```
In [20]: plot randn(1000)
-------> plot(randn(1000))
Out[20]: [<matplotlib.lines.

In [21]: fft randn(1024)
-------> fft(randn(1024))
Out[21]:
array([ 11.94501306 +0.j
        24.187540929 -7.3726
        24.187540929 +7.3726

In [22]: plot fft (randn(102
/usr/lib/pymodules/python2.7
complex values to real disc
  return array(a, dtype, co
-------> plot(fft (randn(102
Out[22]: [<matplotlib.lines.

In [23]: plot fft (randn(102
-------> plot(fft (randn(102
Out[23]: [<matplotlib.lines.

In [24]: 
```

RESEARCH TOOLS 2011
VIDEO 16

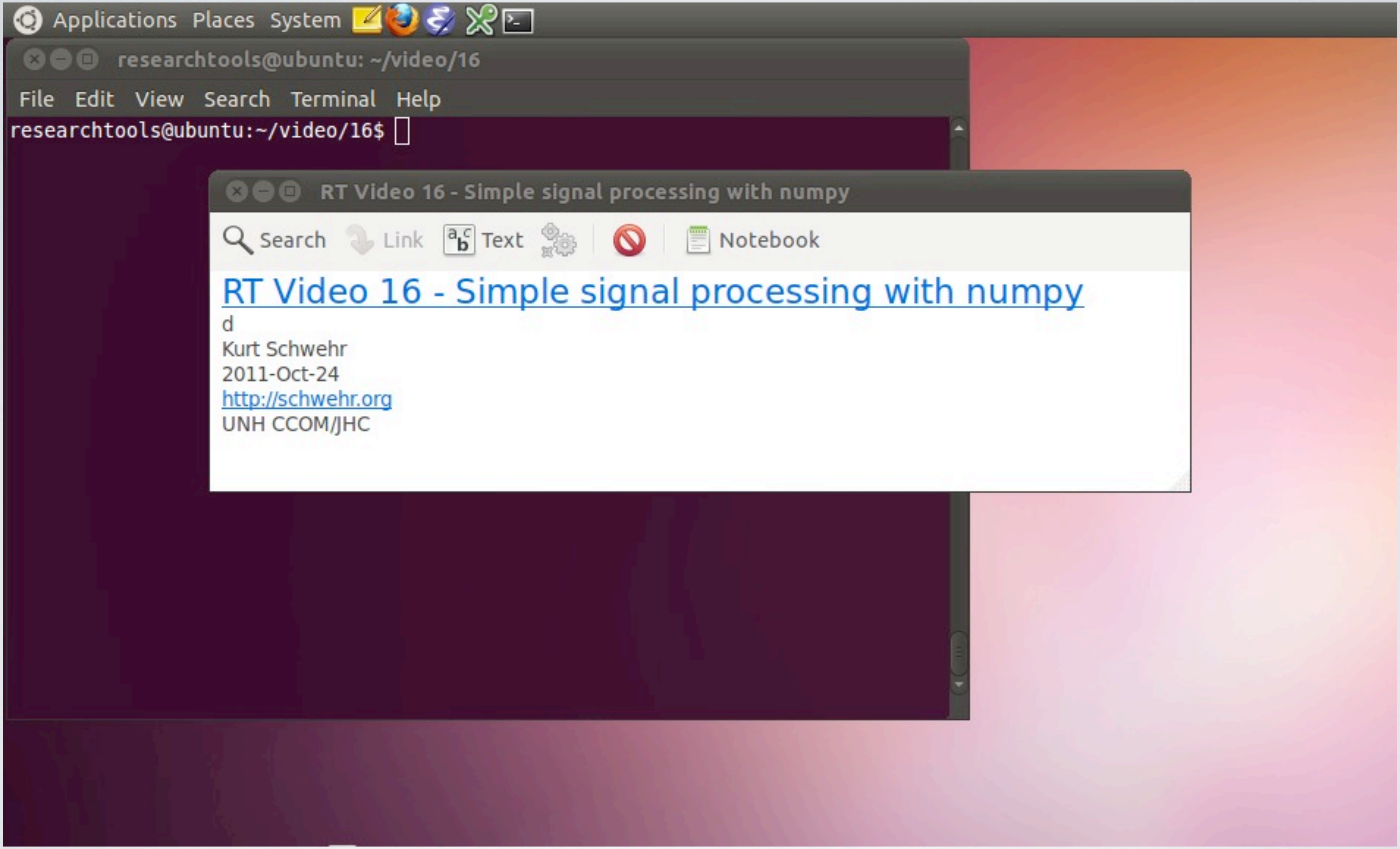
2011-Oct-24
Kurt Schwehr
http://schwehr.org

UNH CCOM/JHC
Python Part 9: Simple signal processing

BY NC SA

http://vislab-ccom.unh.edu/~schwehr/Classes/2011/esci895-researchtools/

http://creativecommons.org/licenses/by-nc-sa/3.0/


Applications
Places
System
researchtools@ubuntu: ~/video/16
File
Edit
View
Search
Terminal
Help
RT Video 16 - Simple signal processing with numpy
Search
Link
Text
Notebook

```
researchtools@ubuntu:~/video/16$
```

RT Video 16 - Simple signal processing with numpy

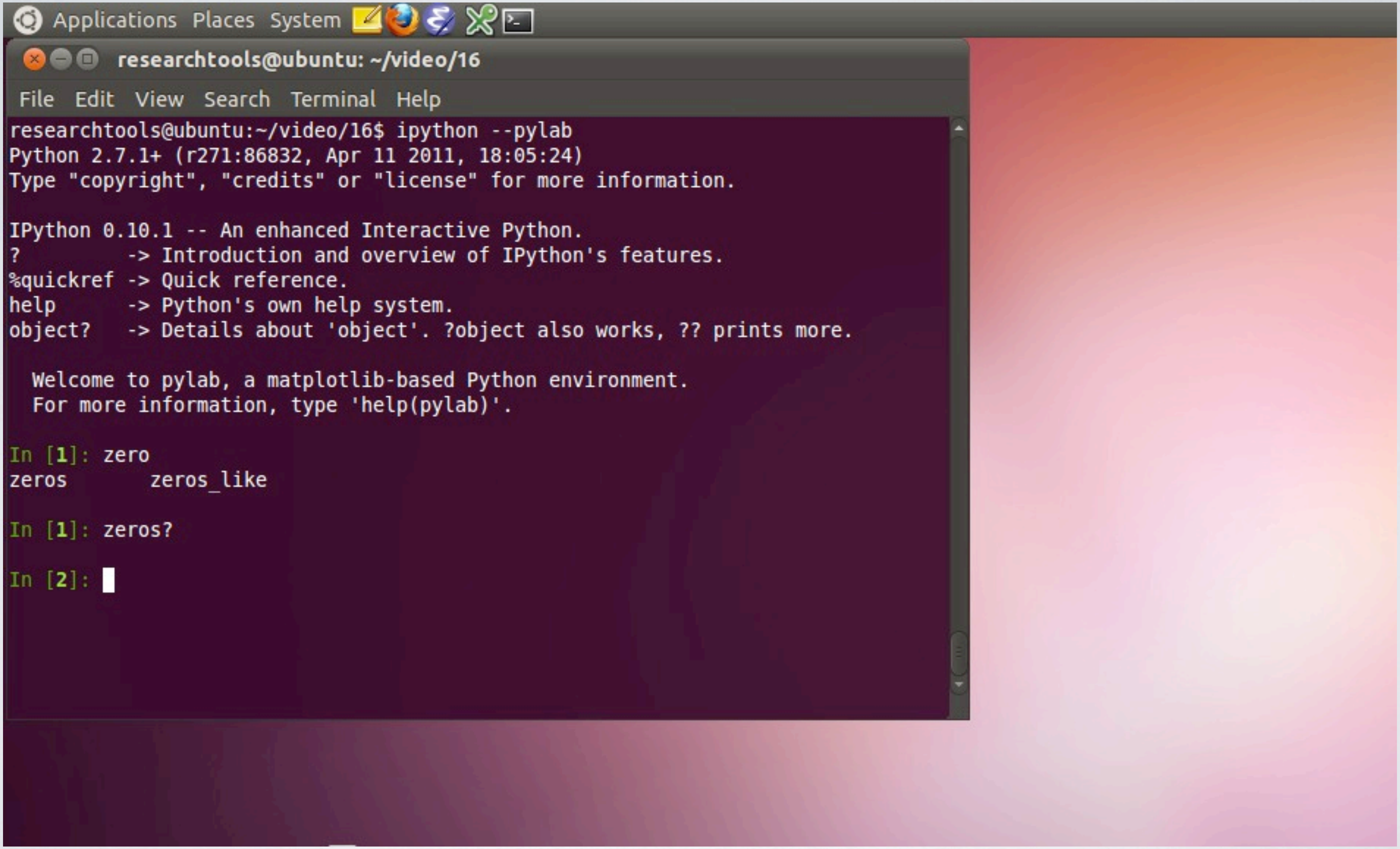
d

Kurt Schwehr

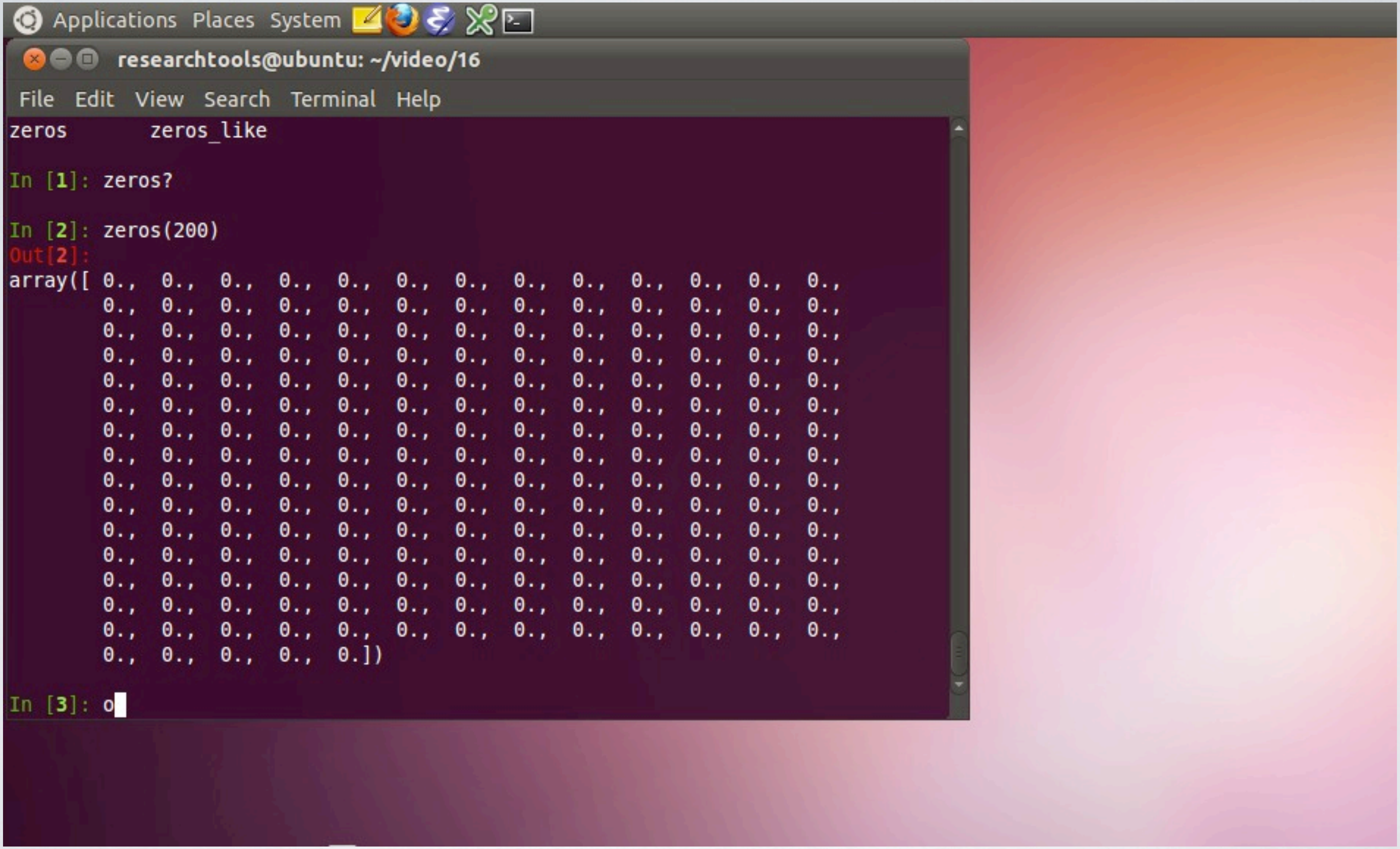
2011-Oct-24

http://schwehr.org

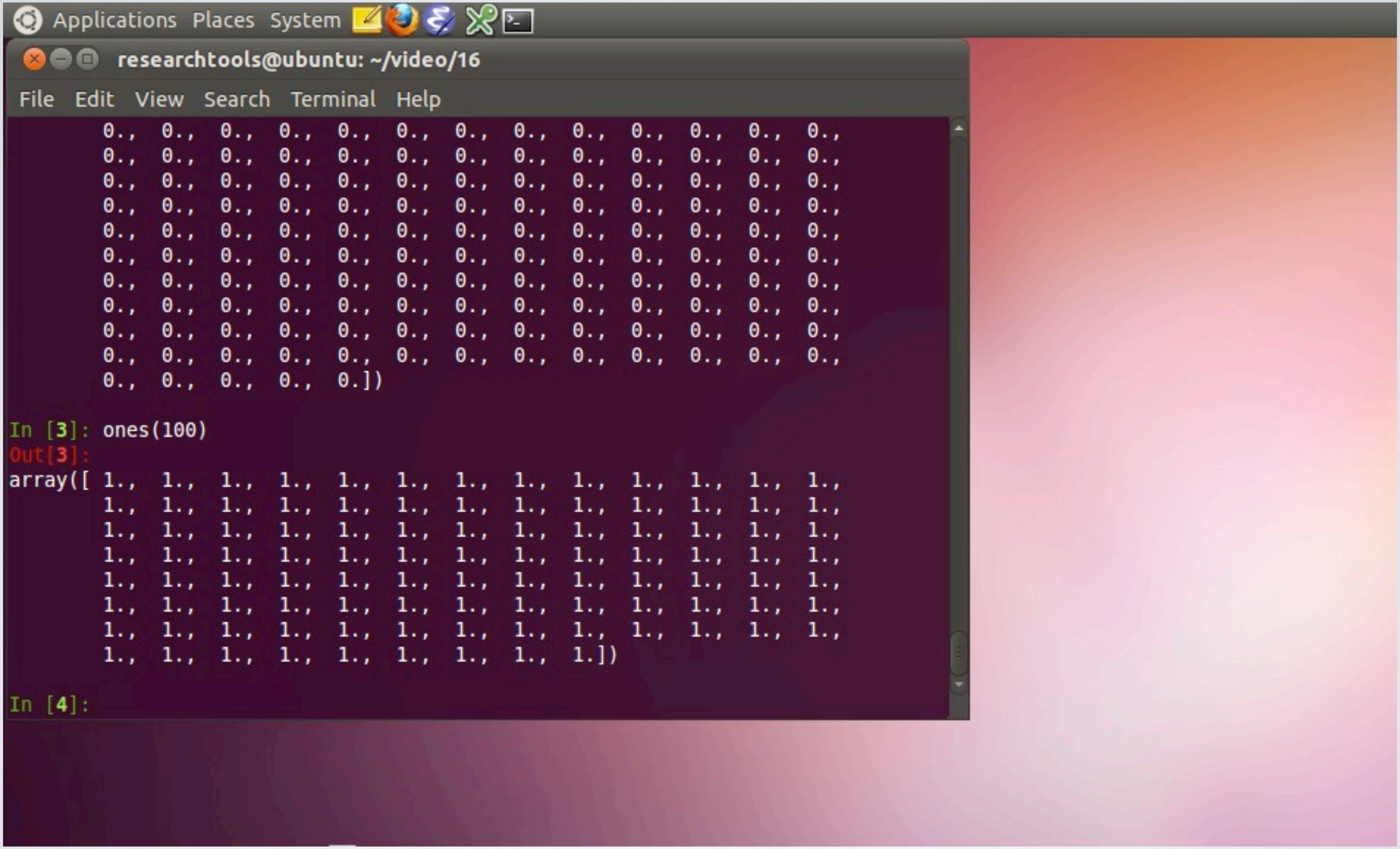
UNH CCOM/JHC



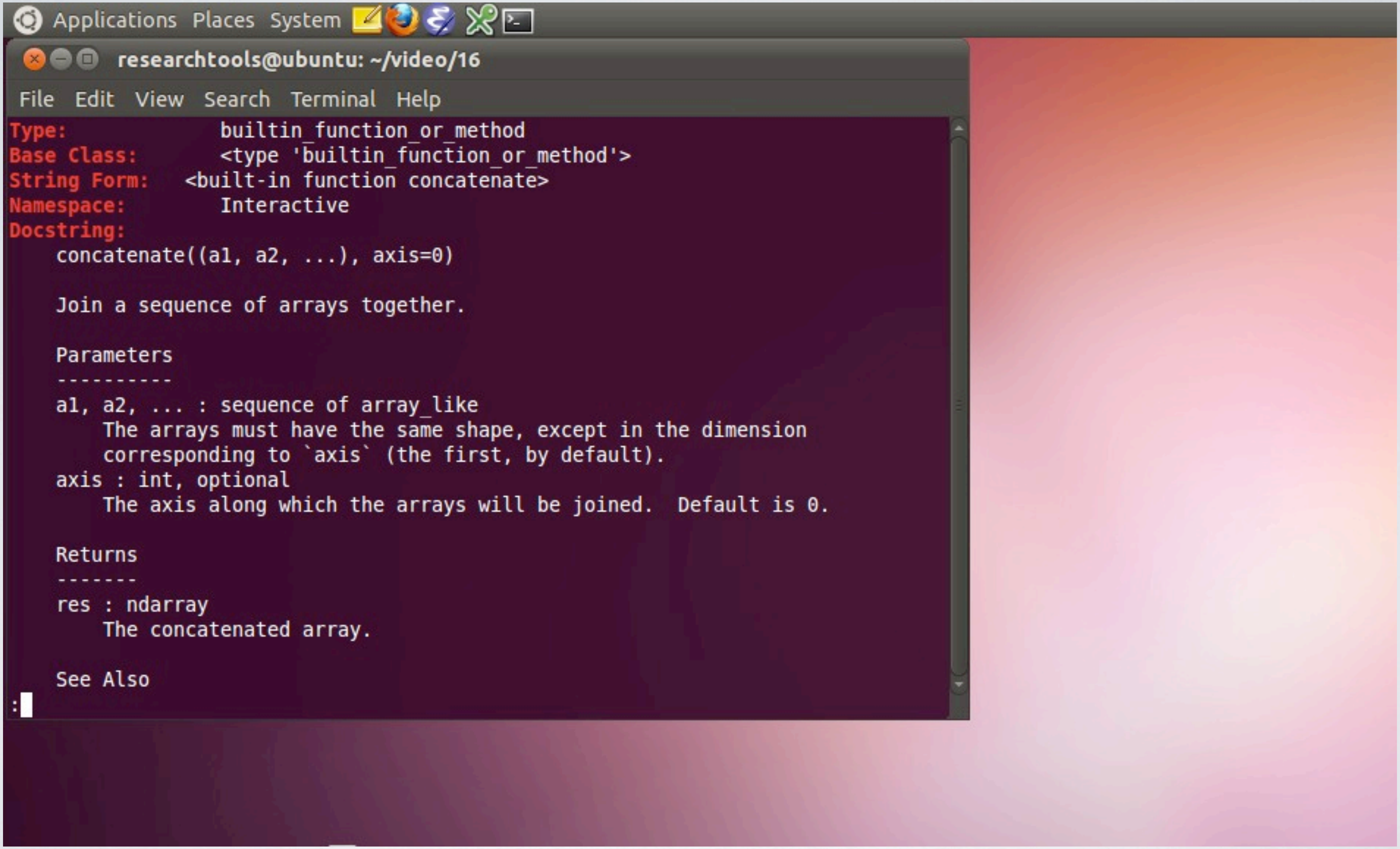
```
researchtools@ubuntu:~/video/16$ ipython --pylab
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
Type "copyright", "credits" or "license" for more information.

IPython 0.10.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object'. ?object also works, ?? prints more.

  Welcome to pylab, a matplotlib-based Python environment.
  For more information, type 'help(pylab)'.

In [1]: zero
zeros       zeros_like

In [1]: zeros?

In [2]: 
```


Applications Places System
researchtools@ubuntu: ~/video/16
File Edit View Search Terminal Help

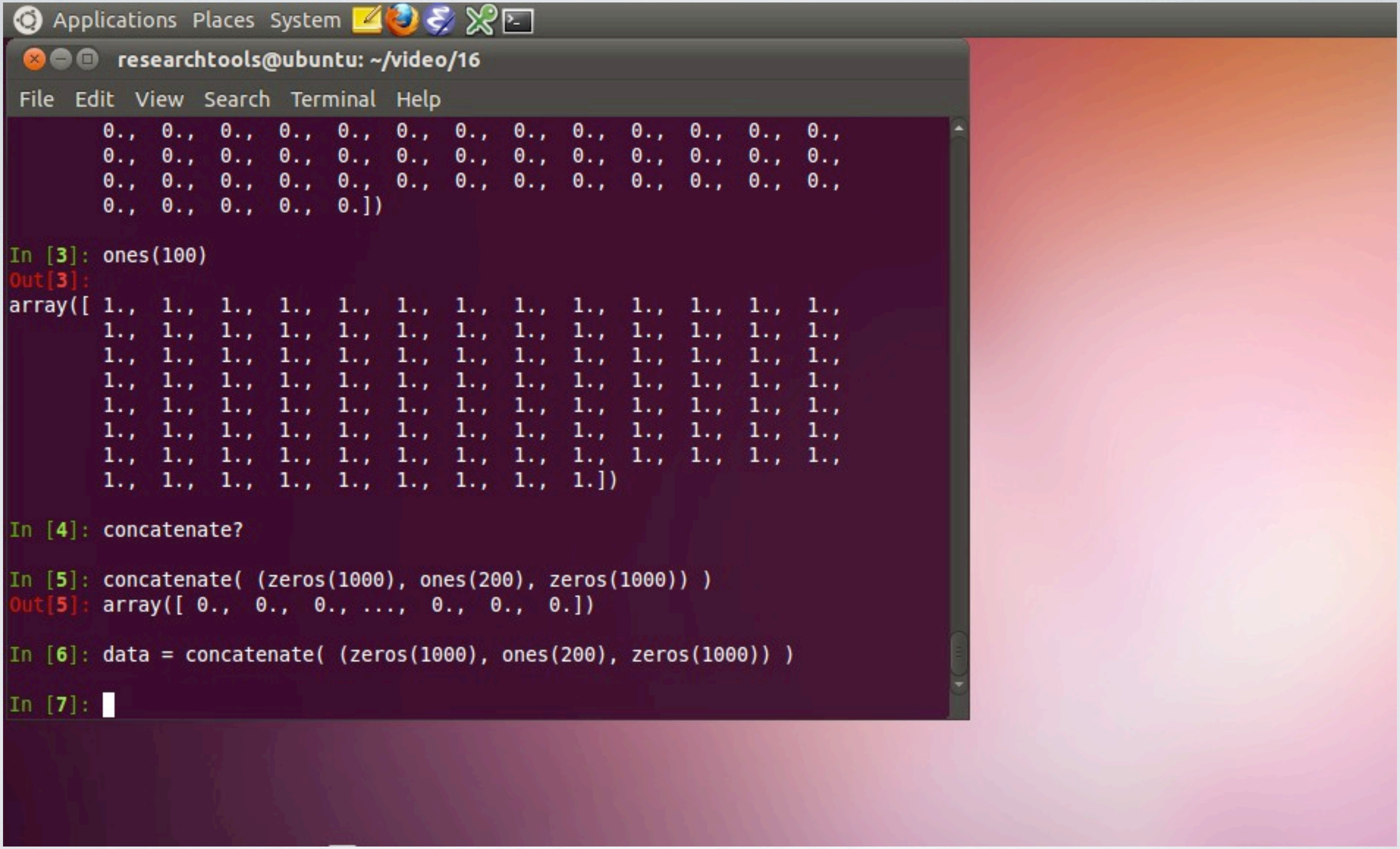
```
zeros       zeros_like

In [1]: zeros?

In [2]: zeros(200)
Out[2]: 
array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.])

In [3]: o
```



```
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.])

In [3]: ones(100)
Out[3]:
array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])

In [4]:
```

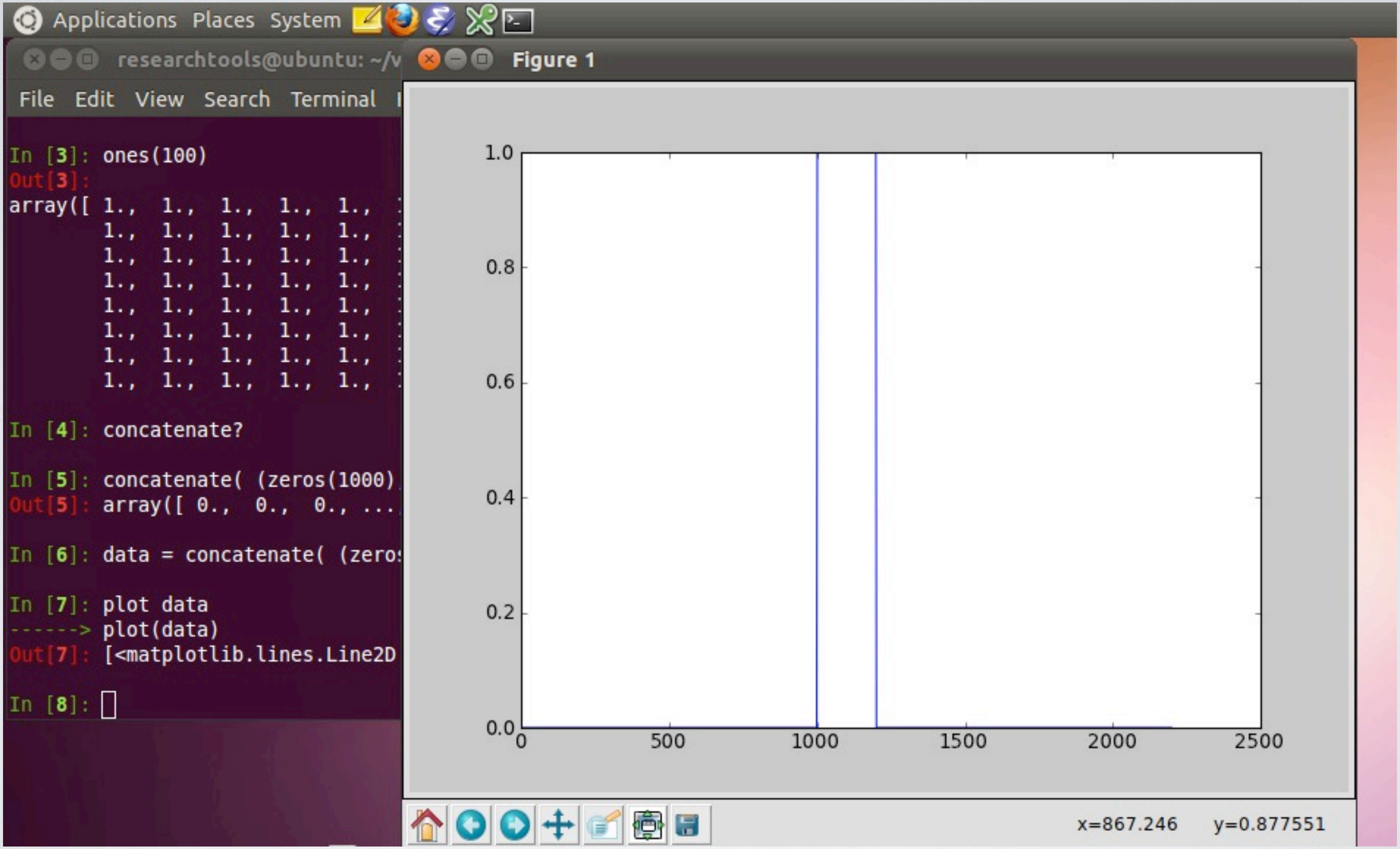


```
Type:             builtin_function_or_method
Base Class:       <type 'builtin_function_or_method'>
String Form:   <built-in function concatenate>
Namespace:        Interactive
Docstring:
    concatenate((a1, a2, ...), axis=0)

    Join a sequence of arrays together.

    Parameters
    ----------
    a1, a2, ... : sequence of array_like
        The arrays must have the same shape, except in the dimension
        corresponding to `axis` (the first, by default).
    axis : int, optional
        The axis along which the arrays will be joined.  Default is 0.

    Returns
    -------
    res : ndarray
        The concatenated array.

    See Also
:
```

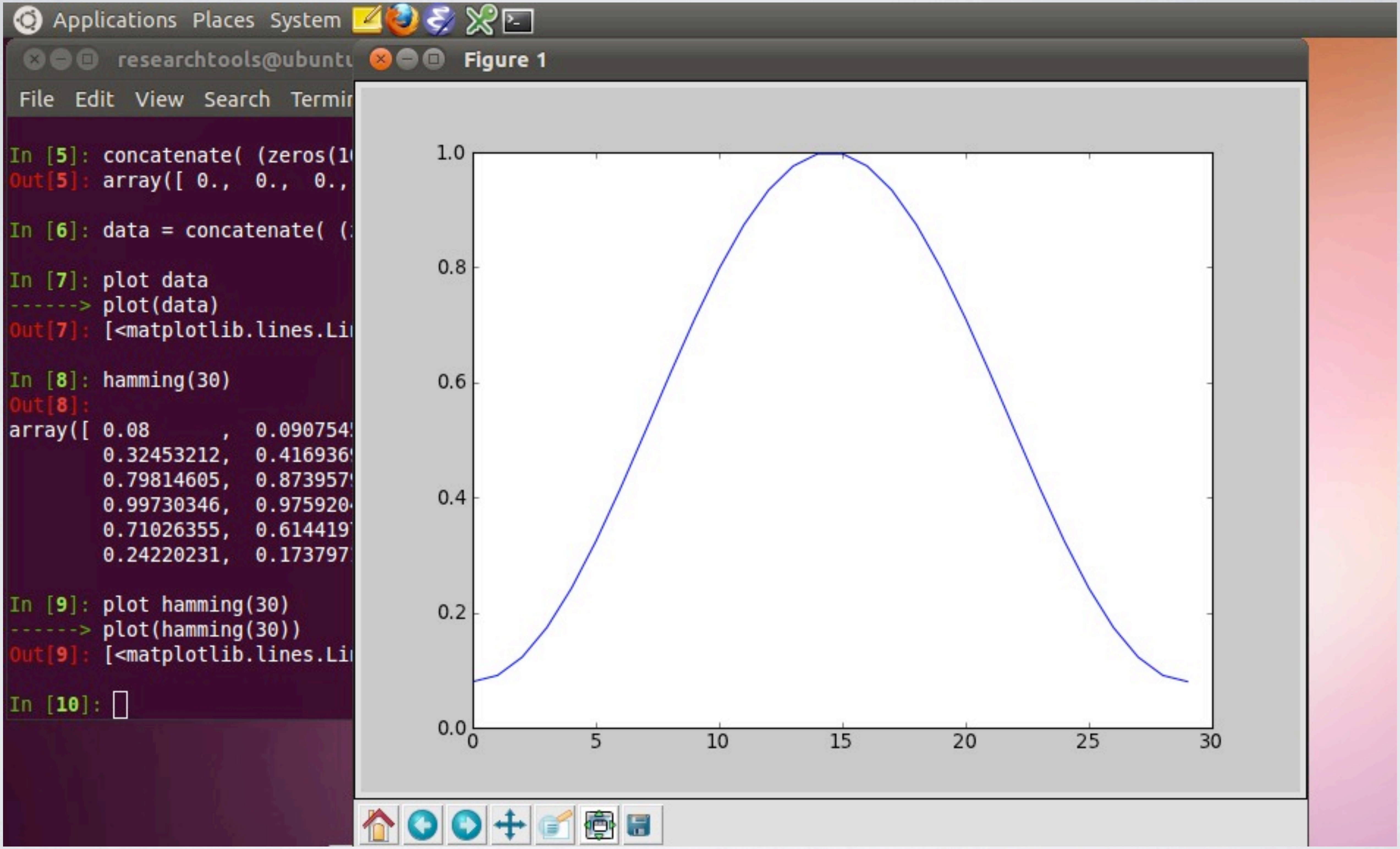


Applications Places System
researchtools@ubuntu: ~/video/16
File Edit View Search Terminal Help

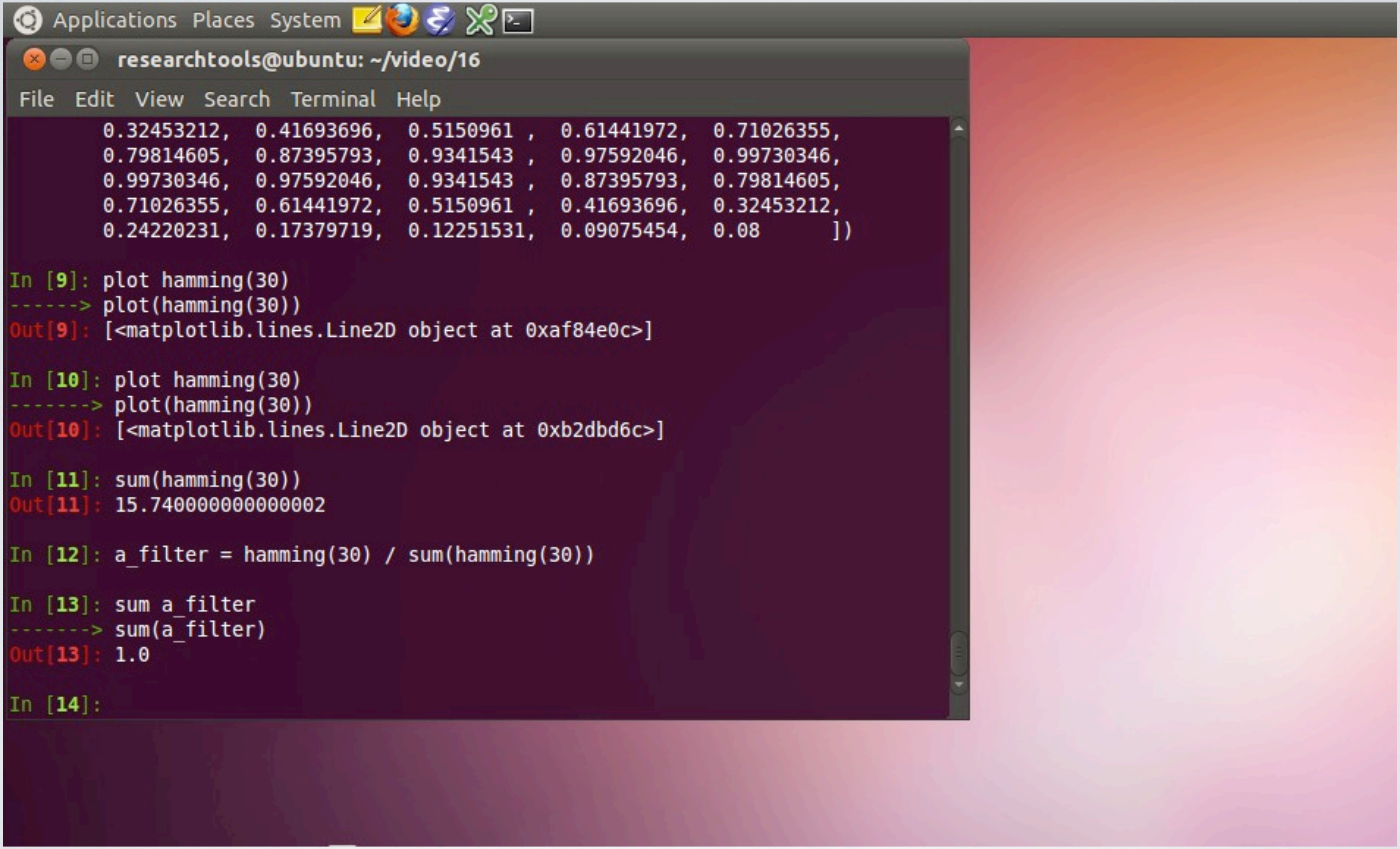
```
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
        0.,  0.,  0.,  0.,  0.])

In [3]: ones(100)
Out[3]:
array([ 1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.])

In [4]: concatenate?

In [5]: concatenate( (zeros(1000), ones(200), zeros(1000)) )
Out[5]: array([ 0.,  0.,  0., ...,  0.,  0.,  0.])

In [6]: data = concatenate( (zeros(1000), ones(200), zeros(1000)) )

In [7]: 
```



```
In [3]: ones(100)
Out[3]:
array([ 1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,
        1.,  1.,  1.,  1.,  1.,

In [4]: concatenate?

In [5]: concatenate( (zeros(1000)
Out[5]: array([ 0.,  0.,  0., ...

In [6]: data = concatenate( (zero

In [7]: plot data
------> plot(data)
Out[7]: [<matplotlib.lines.Line2D

In [8]: 
```



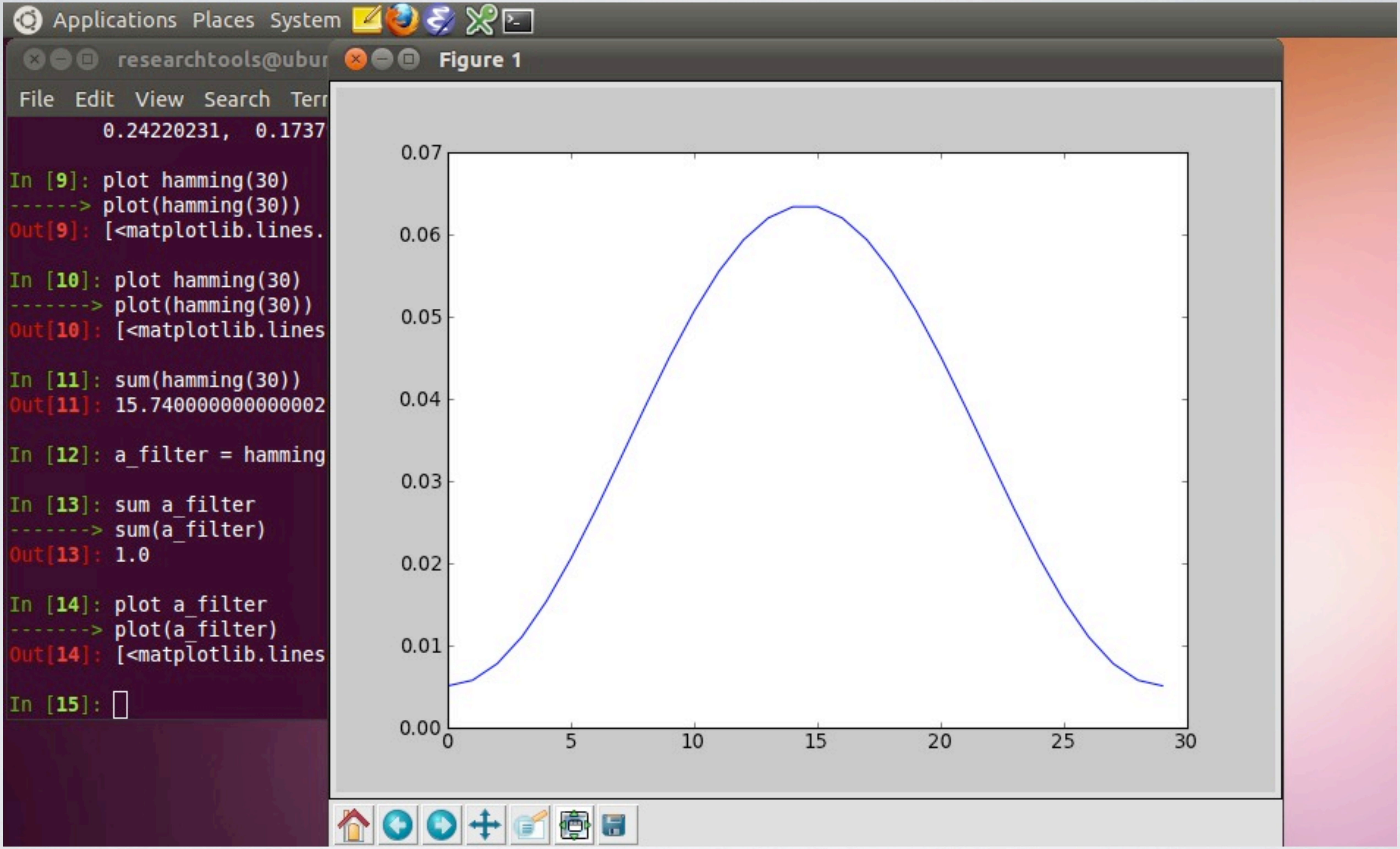
```
In [5]: concatenate( (zeros(1
Out[5]: array([ 0.,  0.,  0.,

In [6]: data = concatenate( (

In [7]: plot data
------> plot(data)
Out[7]: [<matplotlib.lines.Li

In [8]: hamming(30)
Out[8]:
array([ 0.08      ,  0.090754
        0.32453212,  0.416936
        0.79814605,  0.873957
        0.99730346,  0.975920
        0.71026355,  0.614419
        0.24220231,  0.173797

In [9]: plot hamming(30)
------> plot(hamming(30))
Out[9]: [<matplotlib.lines.Li

In [10]: 
```



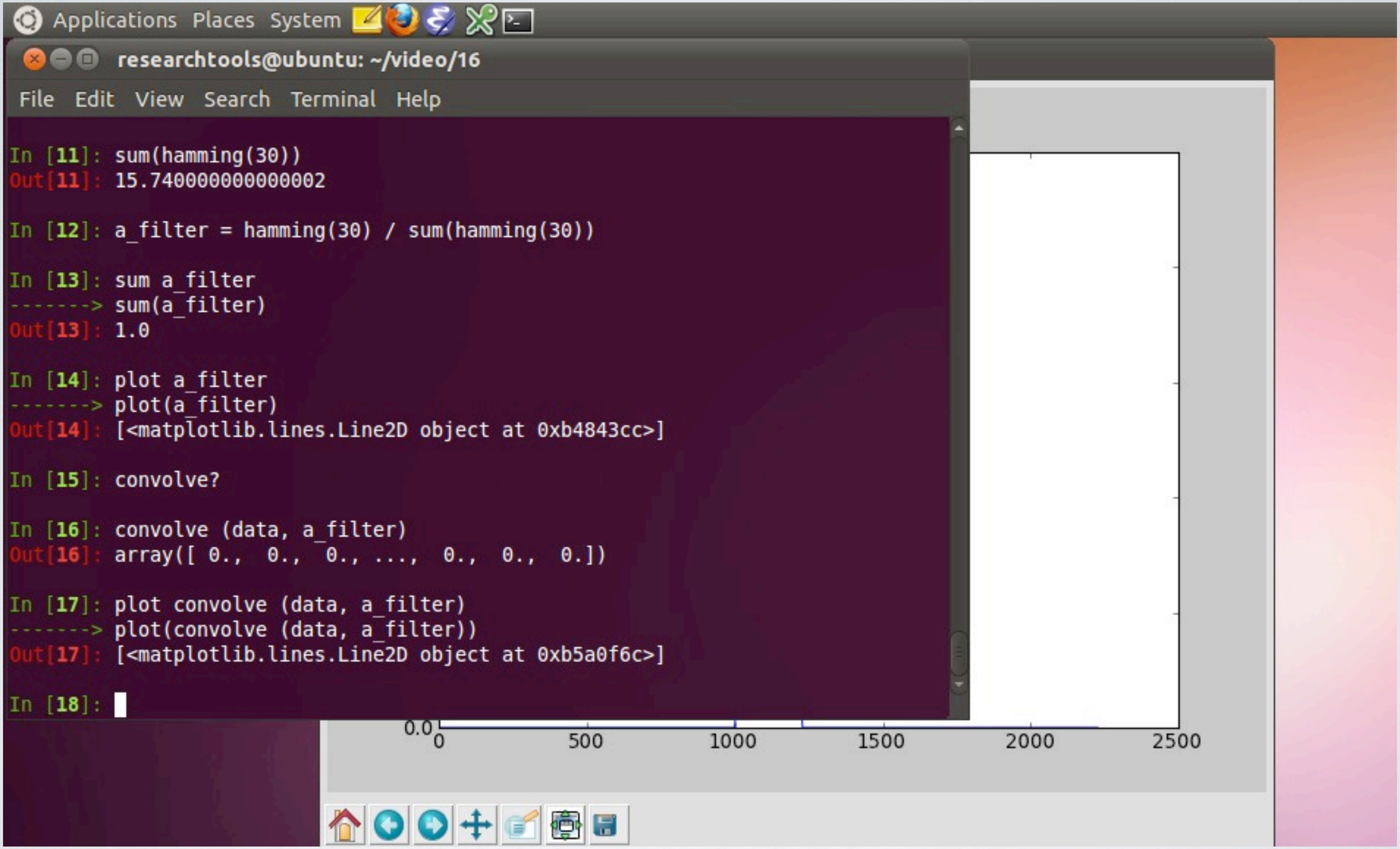
Applications Places System
researchtools@ubuntu: ~/video/16
File Edit View Search Terminal Help

```
        0.32453212,  0.41693696,  0.5150961 ,  0.61441972,  0.71026355,
        0.79814605,  0.87395793,  0.9341543 ,  0.97592046,  0.99730346,
        0.99730346,  0.97592046,  0.9341543 ,  0.87395793,  0.79814605,
        0.71026355,  0.61441972,  0.5150961 ,  0.41693696,  0.32453212,
        0.24220231,  0.17379719,  0.12251531,  0.09075454,  0.08      ])

In [9]: plot hamming(30)
------> plot(hamming(30))
Out[9]: [<matplotlib.lines.Line2D object at 0xaf84e0c>]

In [10]: plot hamming(30)
-------> plot(hamming(30))
Out[10]: [<matplotlib.lines.Line2D object at 0xb2dbd6c>]

In [11]: sum(hamming(30))
Out[11]: 15.740000000000002

In [12]: a_filter = hamming(30) / sum(hamming(30))

In [13]: sum a_filter
-------> sum(a_filter)
Out[13]: 1.0

In [14]:
```

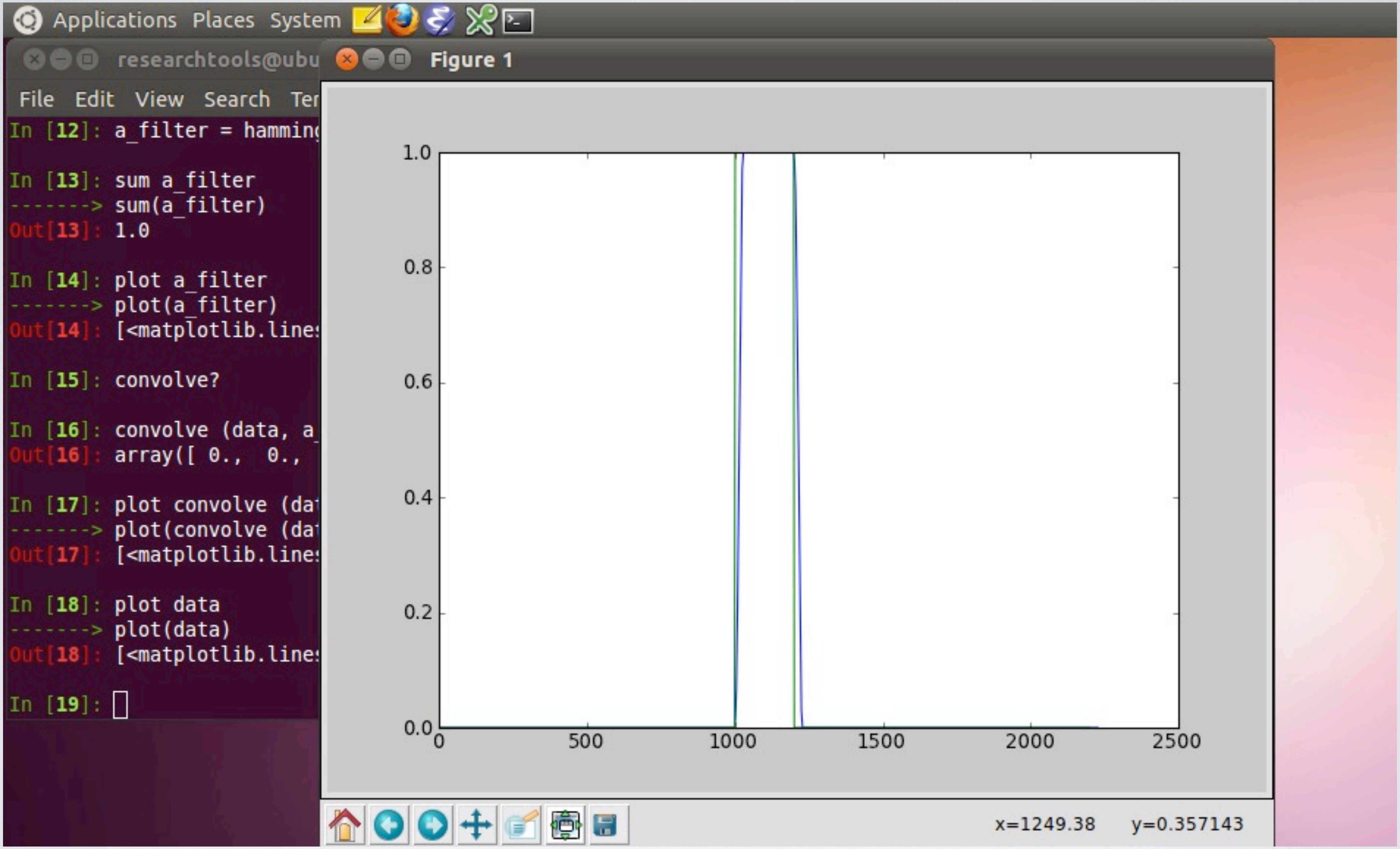
Monday, October 24, 11



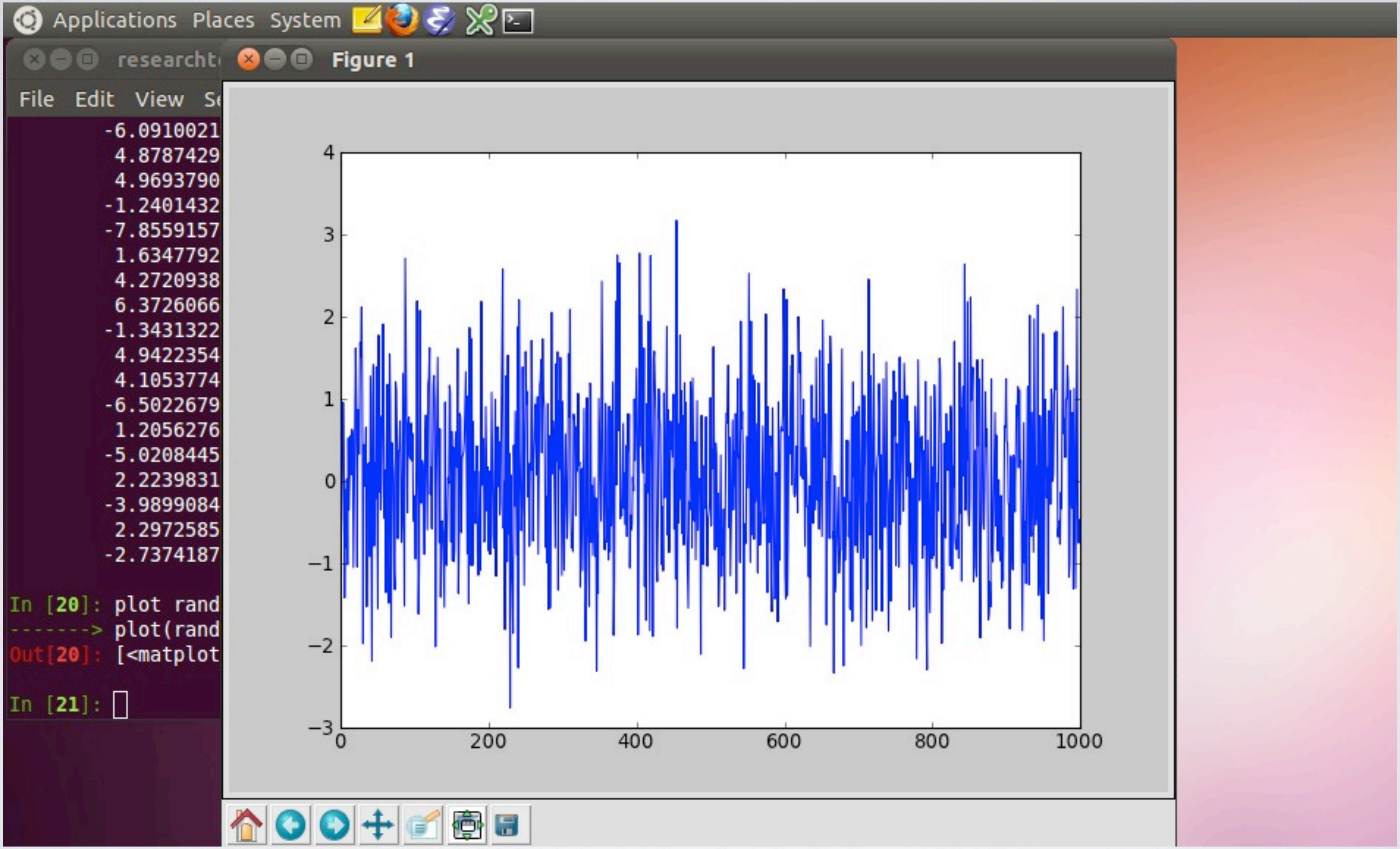
```
       0.24220231,  0.1737

In [9]: plot hamming(30)
------> plot(hamming(30))
Out[9]: [<matplotlib.lines.

In [10]: plot hamming(30)
-------> plot(hamming(30))
Out[10]: [<matplotlib.lines

In [11]: sum(hamming(30))
Out[11]: 15.740000000000002

In [12]: a_filter = hamming

In [13]: sum a_filter
-------> sum(a_filter)
Out[13]: 1.0

In [14]: plot a_filter
-------> plot(a_filter)
Out[14]: [<matplotlib.lines

In [15]: 
```



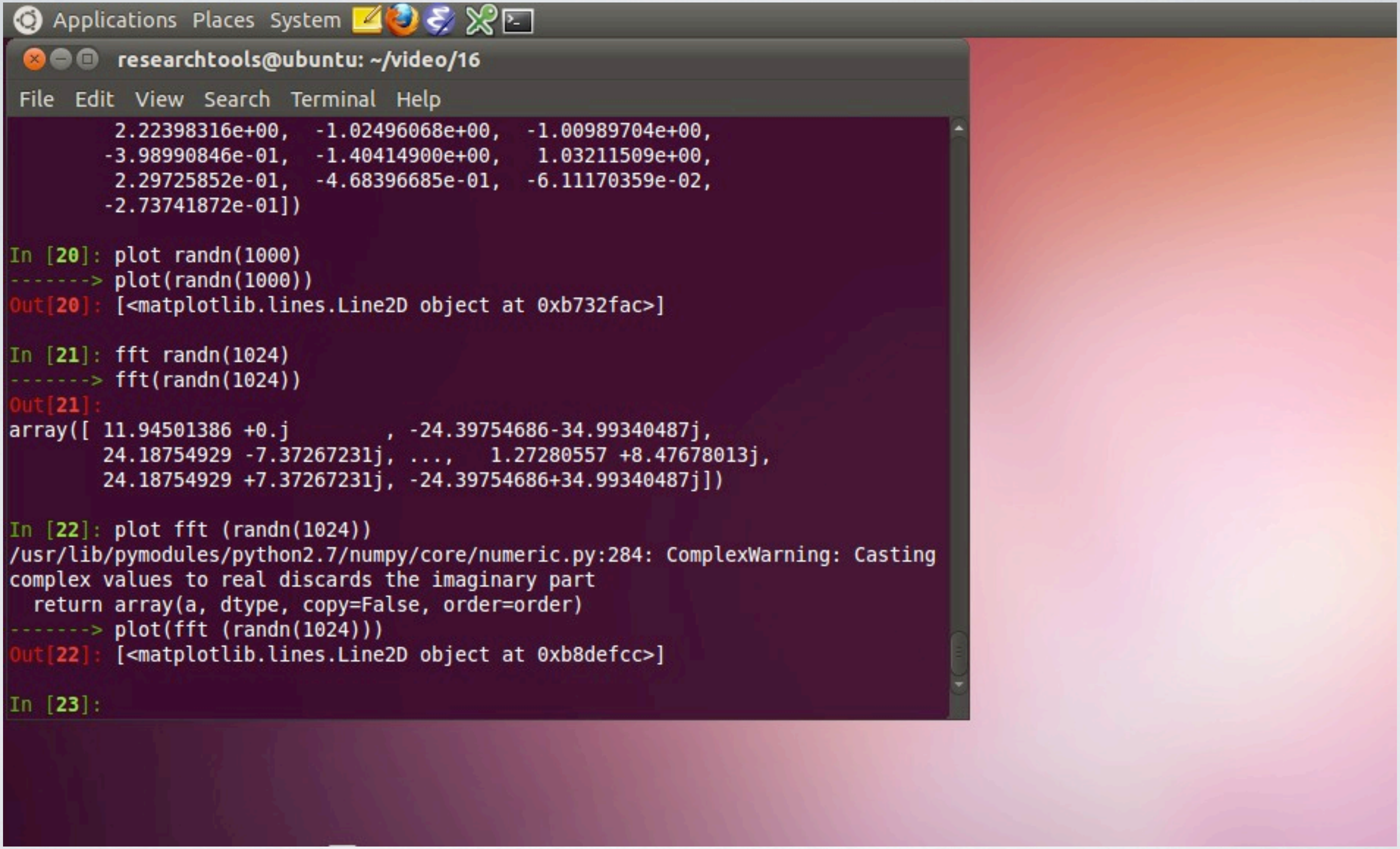
```
In [11]: sum(hamming(30))
Out[11]: 15.740000000000002

In [12]: a_filter = hamming(30) / sum(hamming(30))

In [13]: sum a_filter
-------> sum(a_filter)
Out[13]: 1.0

In [14]: plot a_filter
-------> plot(a_filter)
Out[14]: [<matplotlib.lines.Line2D object at 0xb4843cc>]

In [15]: convolve?

In [16]: convolve (data, a_filter)
Out[16]: array([ 0.,  0.,  0., ...,  0.,  0.,  0.])

In [17]: plot convolve (data, a_filter)
-------> plot(convolve (data, a_filter))
Out[17]: [<matplotlib.lines.Line2D object at 0xb5a0f6c>]

In [18]: 
```



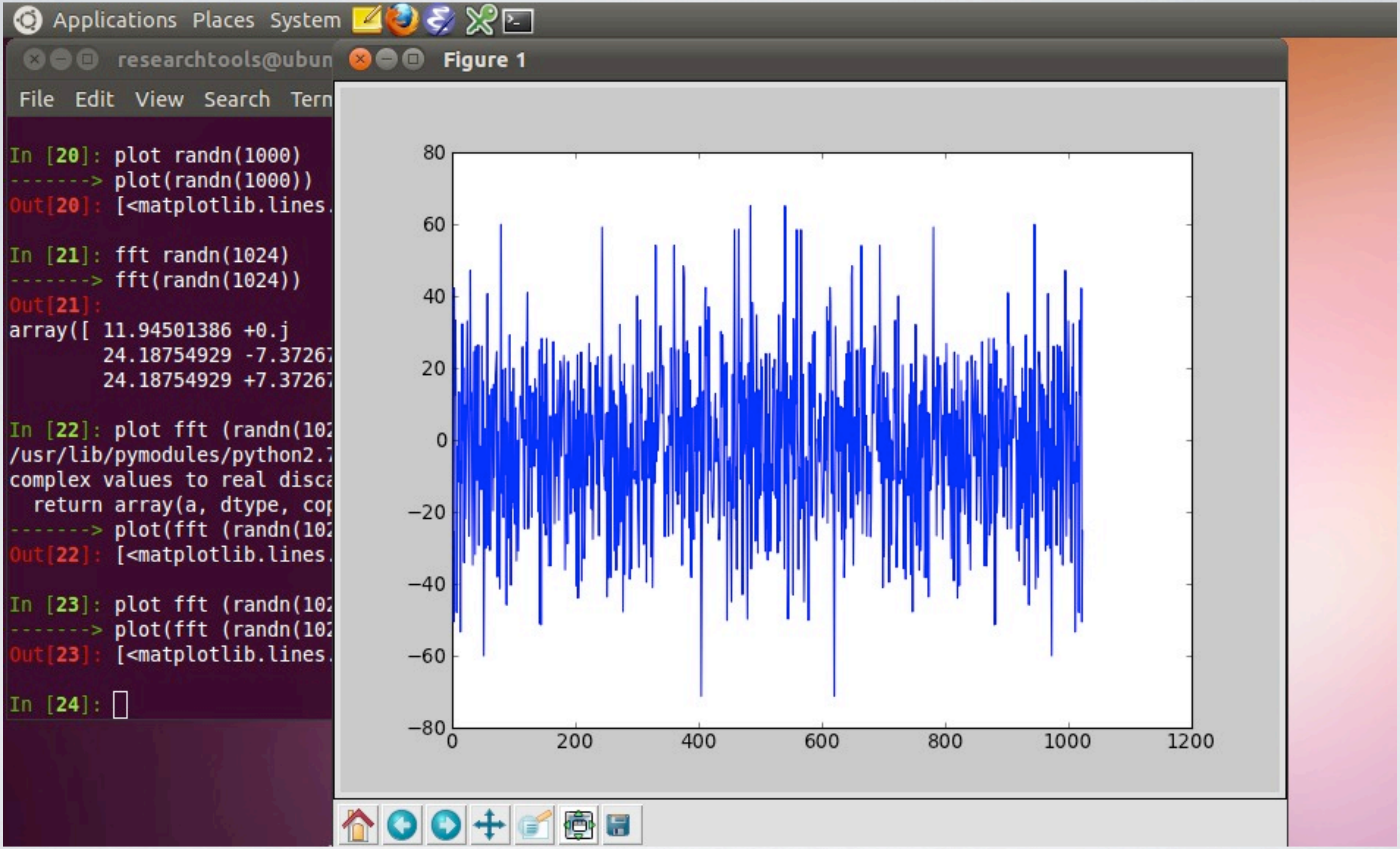
```
In [12]: a_filter = hamming

In [13]: sum a_filter
-------> sum(a_filter)
Out[13]: 1.0

In [14]: plot a_filter
-------> plot(a_filter)
Out[14]: [<matplotlib.line

In [15]: convolve?

In [16]: convolve (data, a_
Out[16]: array([ 0.,  0.,

In [17]: plot convolve (dat
-------> plot(convolve (dat
Out[17]: [<matplotlib.line

In [18]: plot data
-------> plot(data)
Out[18]: [<matplotlib.line

In [19]: 
```



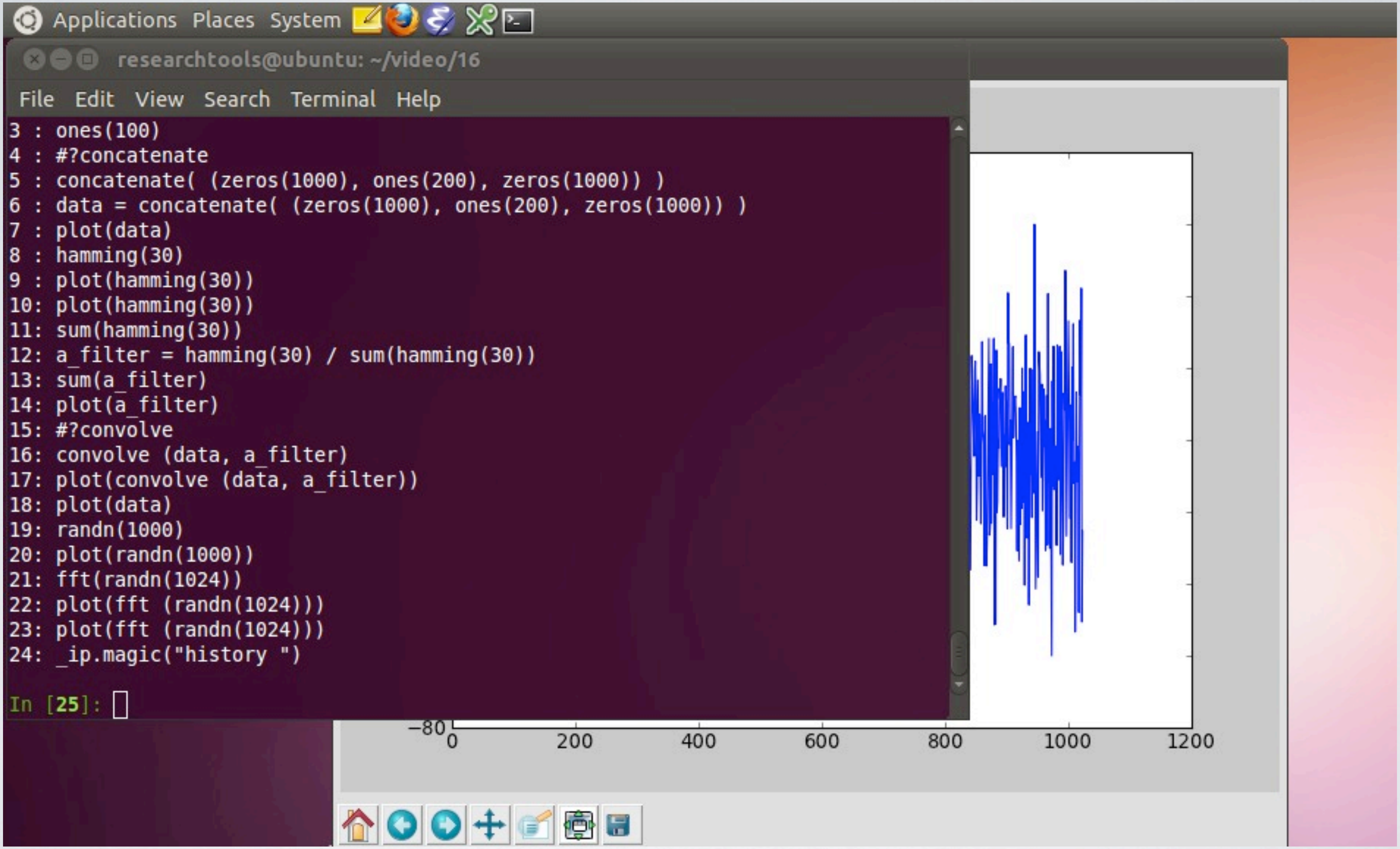
```
         -6.0910021
          4.8787429
          4.9693790
         -1.2401432
         -7.8559157
          1.6347792
          4.2720938
          6.3726066
         -1.3431322
          4.9422354
          4.1053774
         -6.5022679
          1.2056276
         -5.0208445
          2.2239831
         -3.9899084
          2.2972585
         -2.7374187

In [20]: plot rand
-------> plot(rand
Out[20]: [<matplot

In [21]: 
```



```
         2.22398316e+00,  -1.02496068e+00,  -1.00989704e+00,
        -3.98990846e-01,  -1.40414900e+00,   1.03211509e+00,
         2.29725852e-01,  -4.68396685e-01,  -6.11170359e-02,
        -2.73741872e-01])

In [20]: plot randn(1000)
-------> plot(randn(1000))
Out[20]: [<matplotlib.lines.Line2D object at 0xb732fac>]

In [21]: fft randn(1024)
-------> fft(randn(1024))
Out[21]: 
array([ 11.94501386 +0.j        , -24.39754686-34.99340487j,
        24.18754929 -7.37267231j, ...,   1.27280557 +8.47678013j,
        24.18754929 +7.37267231j, -24.39754686+34.99340487j])

In [22]: plot fft (randn(1024))
/usr/lib/pymodules/python2.7/numpy/core/numeric.py:284: ComplexWarning: Casting
complex values to real discards the imaginary part
  return array(a, dtype, copy=False, order=order)
-------> plot(fft (randn(1024)))
Out[22]: [<matplotlib.lines.Line2D object at 0xb8defcc>]

In [23]: 
```



```
In [20]: plot randn(1000)
-------> plot(randn(1000))
Out[20]: [<matplotlib.lines.

In [21]: fft randn(1024)
-------> fft(randn(1024))
Out[21]:
array([ 11.94501386 +0.j
        24.18754929 -7.37267
        24.18754929 +7.37267

In [22]: plot fft (randn(102
/usr/lib/pymodules/python2.7
complex values to real disca
  return array(a, dtype, cop
-------> plot(fft (randn(102
Out[22]: [<matplotlib.lines.

In [23]: plot fft (randn(102
-------> plot(fft (randn(102
Out[23]: [<matplotlib.lines.

In [24]: 
```



```
3 : ones(100)
4 : #?concatenate
5 : concatenate( (zeros(1000), ones(200), zeros(1000)) )
6 : data = concatenate( (zeros(1000), ones(200), zeros(1000)) )
7 : plot(data)
8 : hamming(30)
9 : plot(hamming(30))
10: plot(hamming(30))
11: sum(hamming(30))
12: a_filter = hamming(30) / sum(hamming(30))
13: sum(a_filter)
14: plot(a_filter)
15: #?convolve
16: convolve (data, a_filter)
17: plot(convolve (data, a_filter))
18: plot(data)
19: randn(1000)
20: plot(randn(1000))
21: fft(randn(1024))
22: plot(fft (randn(1024)))
23: plot(fft (randn(1024)))
24: _ip.magic("history ")

In [25]: 
```