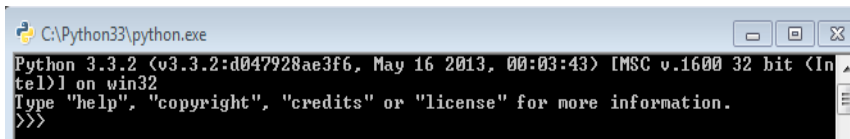
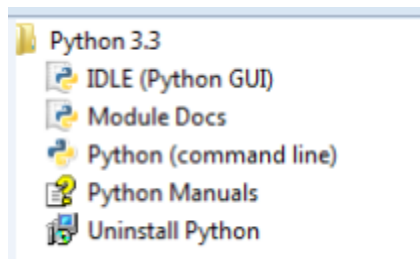
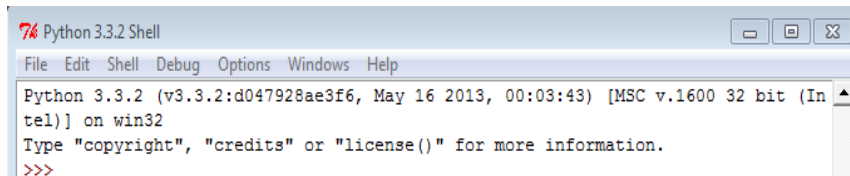
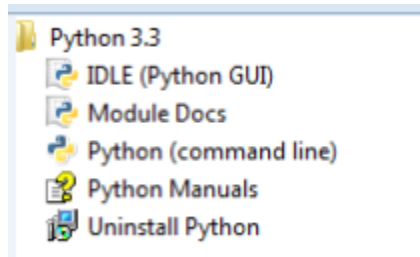


Chapter 1, Introduction and Installation of Python





```
C:\windows\system32\cmd.exe - python
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

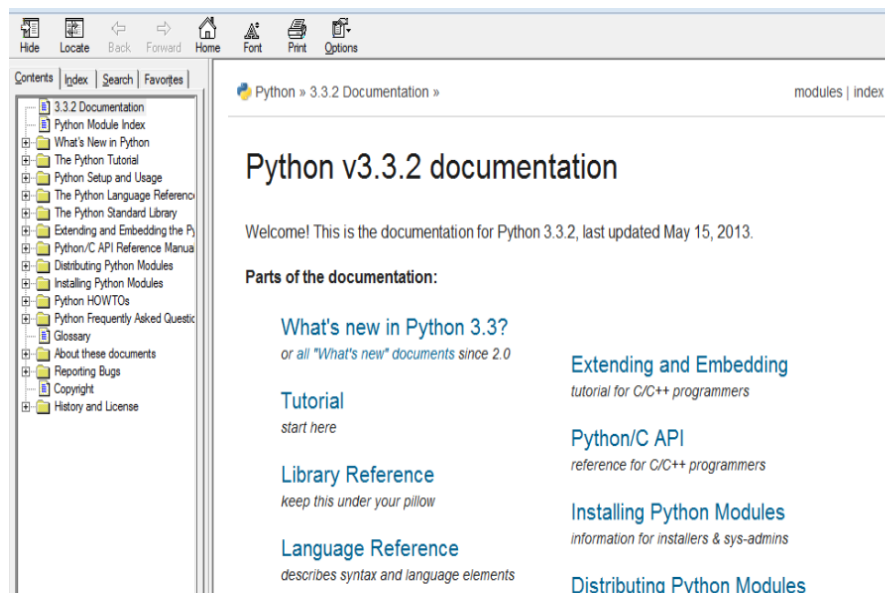
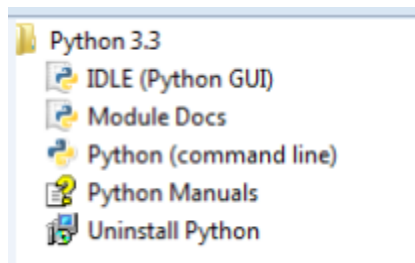
C:\Users\yany>cd \python33
C:\Python33>python
Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013, 00:03:43) [MSC v.1600 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
help> topics

Here is a list of available topics. Enter any topic name to get more help.

ASSERTION          DELETION           LITERALS           SEQUENCES
ASSIGNMENT         DICTIONARIES      LOOPING            SHIFTING
ATTRIBUTEMETHODS  DICTIONARYLITERALS  MAPPINGMETHODS    SLICINGS
ATTRIBUTES         DYNAMICFEATURES   MAPPINGS           SPECIALATTRIBUTES
AUGMENTEDASSIGNMENT ELLIPSIS           METHODS            SPECIALIDENTIFIERS
BASICMETHODS       EXCEPTIONS         MODULES            SPECIALMETHODS
BINARY             EXECUTION          NAMESPACES         STRINGMETHODS
BITWISE            EXPRESSIONS        NONE               STRINGS
BOOLEAN            FILES              NUMBERMETHODS      SUBSCRIPTS
CALLABLEMETHODS    FLOAT              NUMBERS            TRACEBACKS
CALLS              FORMATTING         OBJECTS            TRUTHVALUE
CLASSES            FRAMEOBJECTS       OPERATORS          TUPLELITERALS
CODEOBJECTS        FRAMES             PACKAGES           TUPLES
COMPARISON         FUNCTIONS          POWER              TYPEOBJECTS
COMPLEX            IDENTIFIERS        PRECEDENCE         TYPES
CONDITIONAL        IMPORTING          PRIVATENAMES       UNARY
CONTEXTMANAGERS    INTEGER            RETURNING          UNICODE
CONVERSIONS        LISTLITERALS       SCOPING
DEBUGGING          LISTS              SEQUENCEMETHODS

help>
```



Chapter 3, Using Python as a Financial Calculator

```
>>> from fin101 import *
>>> pv_f(
    (fv, r, n)
    Objective: estimate present value
```

Chapter 4, 13 Lines of Python to Price a Call Option

f _x =NORMDIST(0,0,1,1)			f _x =NORMDIST(-2.3229,0,1,1)			f _x =NORMDIST(1.647,0,1,1)		
D	E	F	D	E	F	D	E	F
0.5			0.010092			0.950221		

Chapter 5, Introduction to Modules

```
>>> dir(__builtins__)
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'Buffer
Error', 'BytesWarning', 'DeprecationWarning', 'EOFError', 'Ellipsis', 'Environme
ntError', 'Exception', 'False', 'FloatingPointError', 'FutureWarning', 'Generato
rExit', 'IOError', 'ImportError', 'ImportWarning', 'IndentationError', 'IndexErr
or', 'KeyError', 'KeyboardInterrupt', 'LookupError', 'MemoryError', 'NameError',
'None', 'NotImplemented', 'NotImplementedError', 'OSError', 'OverflowError', 'P
endingDeprecationWarning', 'ReferenceError', 'RuntimeError', 'RuntimeWarning', 'S
tandardError', 'StopIteration', 'SyntaxError', 'SyntaxWarning', 'SystemError',
'SystemExit', 'TabError', 'True', 'TypeError', 'UnboundLocalError', 'UnicodeDeco
deError', 'UnicodeEncodeError', 'UnicodeError', 'UnicodeTranslateError', 'Unicod
eWarning', 'UserWarning', 'ValueError', 'Warning', 'WindowsError', 'ZeroDivision
Error', '__debug__', '__doc__', '__import__', '__name__', '__package__', 'abs',
'abs', 'all', 'any', 'apply', 'basestring', 'bin', 'bool', 'buffer', 'bytearray', 'a
bytes', 'callable', 'chr', 'classmethod', 'cmp', 'coerce', 'compile', 'complex',
'copyright', 'credits', 'delattr', 'dict', 'dir', 'divmod', 'enumerate', 'eval',
'execfile', 'exit', 'file', 'filter', 'float', 'format', 'frozenset', 'getattr',
'globals', 'hasattr', 'hash', 'help', 'hex', 'id', 'input', 'int', 'intern',
'isinstance', 'issubclass', 'iter', 'len', 'license', 'list', 'locals', 'long',
'map', 'max', 'memoryview', 'min', 'next', 'object', 'oct', 'open', 'ord', 'pow',
'print', 'property', 'quit', 'range', 'raw_input', 'reduce', 'reload', 'repr',
'reversed', 'round', 'set', 'setattr', 'slice', 'sorted', 'staticmethod', 'str',
'sum', 'super', 'tuple', 'type', 'unichr', 'unicode', 'vars', 'xrange', 'zip']
>>>
```

```
>>> import sys as s
>>> s.builtin_module_names
('__builtin__', '__main__', '_ast', '_bisect', '_codecs', '_codecs_cn', '_codecs
_hk', '_codecs_iso2022', '_codecs_jp', '_codecs_kr', '_codecs_tw', '_collections',
'_csv', '_functools', '_heapq', '_hotshot', '_io', '_json', '_locale', '_lspr
of', '_md5', '_multibytecodec', '_random', '_sha', '_sha256', '_sha512', '_sre',
'_struct', '_subprocess', '_syntable', '_warnings', '_weakref', '_winreg', '_arr
ay', '_audioop', '_binascii', '_cPickle', '_cStringIO', '_cmath', '_datetime', '_errno',
'_exceptions', '_future_builtins', '_gc', '_imageop', '_imp', '_itertools', '_marshal',
'_math', '_mmap', '_msvcrt', '_nt', '_operator', '_parser', '_signal', '_strop', '_sys',
'_thread', '_time', '_xxsubtype', '_zipimport', '_zlib')
>>>
```

```
>>> s.modules.keys()
['_heapq', 'functools', 'pyreadline.console.ansi', 'ctypes.os', 'sysconfig', 'logging.os', 'ctypes._endian', 'encodings.encodings', 'pyreadline.modes.basemode', 'logging.stat', 'logging.weakref', 'imp', 'collections', 'logging.thread', 'logging.socket', 'pyreadline.py3k_compat', 'zipimport', 'string', 'encodings.utf_8', 'pyreadline.lineeditor.lineobj', 'logging.logging', 'pyreadline.rlmain', 'signal', 'logging.handlers', 'threading', 'pyreadline.keysyms.common', 'ctypes.wintypes', 'pyreadline.console.console', 'cStringIO', 'logging.threading', 'locale', 'pyreadline.logger', 'pyreadline.lineeditor.wordmatcher', 'atexit', 'pyreadline.unicode_helper', 'encodings', 'logging.traceback', 'pyreadline.console', 'abc', 'ctypes.util', 'pyreadline.modes.notemacs', 're', 'pyreadline.keysyms.keysyms', 'ntpath', 'pyreadline.lineeditor', 'pyreadline.keysyms.winconstants', 'encodings.ascii', 'math', 'UserDict', '_ctypes', 'fnmatch', 'ctypes', 'pyreadline', 'ctypes.struct', 'codecs', 'logging.sys', 'struct', 'functools', '_locale', 'logging', 'socket', 'thread', 'StringIO', 'traceback', 'weakref', 'itertools', 'os', 'future_', 'collections', '_sre', 'pyreadline.lineeditor.history', '__builtin__', 'logging.errno', 'operator', 'logging.re', 'ctypes._ctypes', '_heapq', 'ctypes.sys', 'encodings.cp437', 'errno', '_socket', 'logging.struct', 'sre_constants'
```

```
>>> help()
```

Welcome to Python 2.7! This is the online help utility.

If this is your first time using Python, you should definitely check out the tutorial on the Internet at <http://docs.python.org/2.7/tutorial/>.

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules", "keywords", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose summaries contain a given word such as "spam", type "modules spam".

ArgImagePlugin	_psutil_mswindows	inspect	sklearn
BaseHTTPServer	_pyio	io	smtplib
Bastion	_pytest	isapi	smtplib
BdfFontFile	_random	itertools	sndhdr
Bio	_sha	itsdangerous	socket
BioSQL	_sha256	jinjja2	sphinx
BmpImagePlugin	_sha512	json	spyderlib
BufRStubImagePlugin	_socket	keyring	spyderplugins
CGIHTTPServer	_sqlite3	keyword	sqlalchemy
Canvas	_sre	kiva	sqlite3
ConfigParser	_ssl	launcher	sre
ContainerIO	_strptime	lib2to3	sre_compile
Cookie	_struct	linecache	sre_constants
Crypto	_subprocess	llpython	sre_parse
CurImagePlugin	_symtable	llvm	ssl
Cython	_system_path	llvm_array	sspi
DcxImagePlugin	_testcapi	llvm_builder	sspicon
Dialog	_threading_local	llvmmath	stat
DocXMLRPCServer	_tkinter	llumpy	statsmodels
EpsImagePlugin	_warnings	locale	statvfs
ExifTags	_weakref	logging	storemagic
FileDialog	_weakrefset	lxml	string

```

_license      httplib      setuptools   xmllib
_locale      httplib      sgmlib       xmlrpclib
_lsprof      idlelib      sha          xxsubtype
_markerlib   ihooks      shelve       yaml
_md5         imageop     shlex        zipfile
_msi         imaplib     shutil       zipimport
_multibytecode imghdr      signal       zlib
_multiprocessing imp         site         zmq
_nsis        importlib   six
_osx_support imputil     skimage

```

Enter any module name to get more help. Or, type "modules spam" to search for modules whose descriptions contain the word "spam".

help>

```

help> modules matplotlib

```

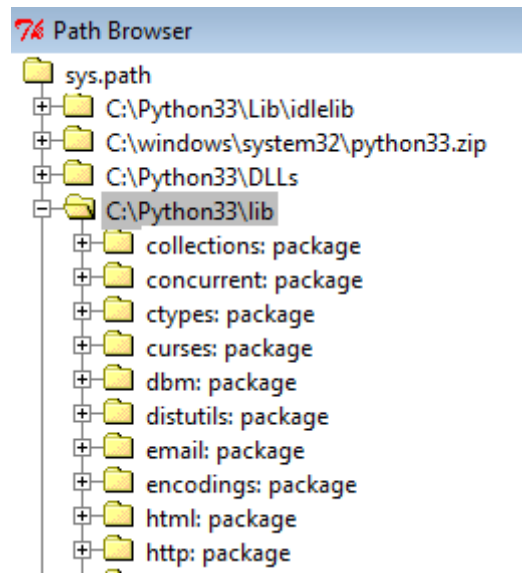
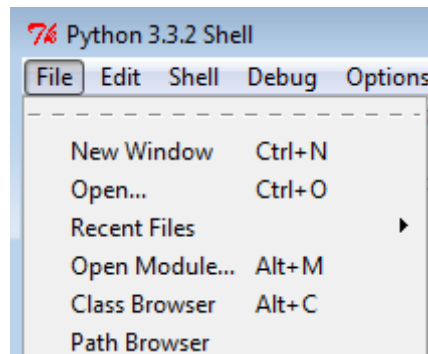
Here is a list of matching modules. Enter any module name to get more help.

```

IPython.core.magics.pylab - Implementation of magic functions for matplotlib/pylab support.
IPython.core.pylabtools - Pylab (matplotlib) support utilities.
bokeh.plotting - Command-line driven plotting functions, a la Matplotlib / Matlab / etc.
matplotlib - This is an object-oriented plotting library.
matplotlib._cm - Nothing here but dictionaries for generating LinearSegmentedColormaps.
matplotlib._cntr
matplotlib._de launay
matplotlib._image
matplotlib._math text_data - font data tables for truetype and afm computer modern fonts
matplotlib._path
matplotlib._png
matplotlib._pylab_helpers - Manage figures for pyplot interface.
matplotlib._tri
matplotlib._windowing

```

- Python 3.3
- IDLE (Python GUI)
 - Module Docs
 - Python (command line)
 - Python Manuals
 - Uninstall Python



Chapter 6, Introduction to NumPy and SciPy

```
python.exe - Shortcut
Python 2.7.5 |Anaconda 1.7.0 (32-bit)| (default, Jul 1 2013, 12:41:55) [MSC v.1
500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

fx =NPV(0.1,E1:I1)					
D	E	F	G	H	I
-100	50	40	20	10	50
\$131.41					

fx =PMT(0.045/12,30*12,250000)		
D	E	F
(\$1,266.71)		

```
>>> dir(np)
['ALLOW_THREADS', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'DataSource', 'ERR_CALL',
'ERR_DEFAULT', 'ERR_DEFAULT2', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_RAISE',
'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVALID', 'FPE_
OVERFLOW', 'FPE_UNDERFLOW', 'False', 'Inf', 'Infinity', 'MAXDIMS', 'MachAr',
'NaN', 'NINF', 'NZERO', 'NaN', 'PINF', 'PZERO', 'PackageLoader', 'RAISE', 'RankWa
rning', 'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDERFLC
W', 'ScalarType', 'Tester', 'True', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_PYVALS_NAME',
'WRAP', '_NUMPY_SETUP_', '_all_', '_builtins_', '_config_', '_doc_',
'_file_', '_git_revision_', '_name_', '_package_', '_path_', '_vers
ion_', '_import_tools_', '_mat_', 'abs', 'absolute', 'add', 'add_docstring', 'add
```

```
>>> x[200:210]
array(['equal', 'errstate', 'exp', 'exp2', 'expand_dims', 'expm1',
'extract', 'eye', 'fabs', 'fastCopyAndTranspose', 'l',
dtype='!S22'])
>>> x[200:250]
array(['equal', 'errstate', 'exp', 'exp2', 'expand_dims', 'expm1',
'extract', 'eye', 'fabs', 'fastCopyAndTranspose', 'fft',
'fill_diagonal', 'find_common_type', 'find_info', 'fix', 'flatiter',
'flatnonzero', 'flexible', 'fliplr', 'flipud', 'float', 'float16',
'float32', 'float64', 'float_', 'floating', 'floor', 'floor_divide',
'fmax', 'fmin', 'fmod', 'format_parser', 'frexp', 'frombuffer',
'fromfile', 'fromfunction', 'fromiter', 'frompyfunc', 'fromregex',
'fromstring', 'fv', 'generic', 'genfromtxt', 'get_array_wrap',
'get_include', 'get_numarray_include', 'get_printoptions',
'getbuffer', 'getbufsize', 'geterr', 'l',
dtype='!S22'])
>>>
```



```
>>> dir(sp)
['ALLOW_THREADS', 'BUFSIZE', 'CLIP', 'ComplexWarning', 'DataSource', 'ERR_CALL',
 'ERR_DEFAULT', 'ERR_DEFAULT2', 'ERR_IGNORE', 'ERR_LOG', 'ERR_PRINT', 'ERR_RAISE',
 'ERR_WARN', 'FLOATING_POINT_SUPPORT', 'FPE_DIVIDEBYZERO', 'FPE_INVALID', 'FPE_
OVERFLOW', 'FPE_UNDERFLOW', 'False_', 'Inf', 'Infinity', 'MAXDIMS', 'MachAr', '
NAN', 'NINF', 'NZERO', 'NaN', 'PIPF', 'PZERO', 'PackageLoader', 'RAISE', 'RankWa
rning', 'SHIFT_DIVIDEBYZERO', 'SHIFT_INVALID', 'SHIFT_OVERFLOW', 'SHIFT_UNDERFLO
W', 'ScalarType', 'Tester', 'True_', 'UFUNC_BUFSIZE_DEFAULT', 'UFUNC_PYVALS_NAME',
 '__file__', '__name__', '__numpy_version__', '__package__', '__path__', '__ver
sion__', 'absolute', 'absolute_import', 'add', 'add_docstring', 'add_newdoc', 'a
dd_newdoc_ufunc', 'add_newdocs', 'alen', 'all', 'allclose', 'alltrue', 'alterdot',
 'amax', 'amin', 'angle', 'any', 'append', 'apply_along_axis', 'apply_over_axe
s', 'arange', 'arccos', 'arccosh', 'arcsin', 'arcsinh', 'arctan', 'arctan2', 'ar
ctanh', 'argmax', 'argmin', 'argsort', 'argwhere', 'around', 'array', 'array2str
ing', 'array_equal', 'array_equiv', 'array_repr', 'array_split', 'array_str', 'a
```

```
>>> import numpy as np
>>> help(np.std)
Help on function std in module numpy.core.fromnumeric:

std(a, axis=None, dtype=None, out=None, ddof=0, keepdims=False)
    Compute the standard deviation along the specified axis.

    Returns the standard deviation, a measure of the spread of a distribution,
    of the array elements. The standard deviation is computed for the
    flattened array by default, otherwise over the specified axis.

    Parameters
    -----
    a : array_like
        Calculate the standard deviation of these values.
    axis : int, optional
        Axis along which the standard deviation is computed. The default is
        to compute the standard deviation of the flattened array.
```

```
python.exe - Shortcut
Help on package numpy:

NAME
    numpy

FILE
    c:\anaconda\lib\site-packages\numpy\__init__.py

DESCRIPTION
    NumPy
    =====

    Provides
    1. An array object of arbitrary homogeneous items
    2. Fast mathematical operations over arrays
    3. Linear Algebra, Fourier Transforms, Random Number Generation

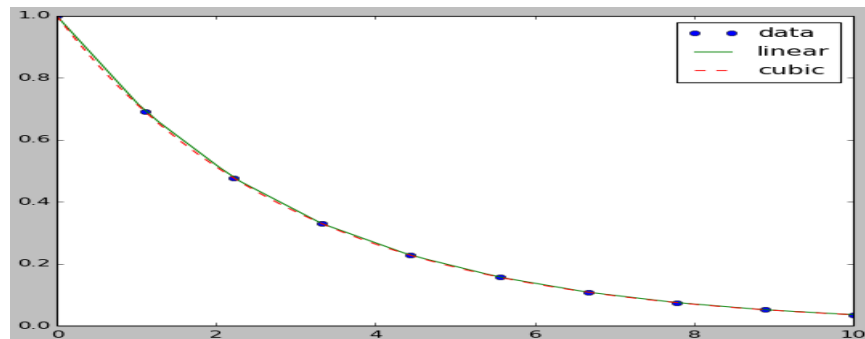
    How to use the documentation
    -----
    Documentation is available in two forms: docstrings provided
    with the code, and a loose standing reference guide, available from
    `the NumPy homepage <http://www.scipy.org>`.

    We recommend exploring the docstrings using
    -- More --
```

```

>>> from scipy import stats
>>> dir(stats)
['Tester', '__all__', '__builtins__', '__doc__', '__file__', '__name__', '__pack
age__', '__path__', 'binned_statistic', 'rank', 'support', 'tukeylambda_stat
s', 'absolute_import', 'alpha', 'anderson', 'anglit', 'ansari', 'arcsine', 'bart
lett', 'bayes_mv', 'bernoulli', 'beta', 'betai', 'betaprime', 'binned_statistic
', 'binned_statistic_2d', 'binned_statistic_dd', 'binom', 'binom_test', 'boltzma
nn', 'boxcox', 'boxcox_llf', 'boxcox_normmax', 'boxcox_normplot', 'bradford', 'b
urr', 'callable', 'cauchy', 'chi', 'chi2', 'chi2_contingency', 'chisqprob', 'chi
square', 'circmean', 'circstd', 'circvar', 'cmedian', 'contingency', 'cosine', 'c
umfreq', 'describe', 'dgamma', 'distributions', 'division', 'dlaplace', 'dweibu
ll', 'entropy', 'erlang', 'expon', 'exponpow', 'exponweib', 'f', 'f_oneway', 'f_
value', 'f_value_multivariate', 'f_value_wilks_lambda', 'fastsort', 'fatiguelife
', 'find_repeats', 'fisher_exact', 'fisk', 'fligner', 'foldcauchy', 'foldnorm',
'fprob', 'frechet_l', 'frechet_r', 'friedmanchisquare', 'futil', 'gamma', 'gauss
hyper', 'gaussian_kde', 'genexpon', 'genextreme', 'gengamma', 'genhalflogistic',
'genlogistic', 'genpareto', 'geom', 'gilbrat', 'glm', 'gmean', 'gompertz', 'gum
bel_l', 'gumbel_r', 'halfcauchy', 'halflogistic', 'halfnorm', 'histogram', 'hist

```



```

Univariate distributions
=====
beta          Beta distribution over ``[0, 1]``.
binomial      Binomial distribution.
chisquare     :math:`\chi^2` distribution.
exponential   Exponential distribution.
f            F (Fisher-Snedecor) distribution.
gamma        Gamma distribution.
geometric    Geometric distribution.
gumbel       Gumbel distribution.
hypergeometric Hypergeometric distribution.
laplace      Laplace distribution.
logistic     Logistic distribution.
lognormal    Log-normal distribution.
logseries    Logarithmic series distribution.
negative_binomial Negative binomial distribution.
noncentral_chisquare Non-central chi-square distribution.
noncentral_f Non-central F distribution.
normal       Normal / Gaussian distribution.
pareto       Pareto distribution.
poisson      Poisson distribution.
power        Power distribution.
rayleigh     Rayleigh distribution.
triangular   Triangular distribution.
uniform      Uniform distribution.
vonmises     Von Mises circular distribution.
wald         Wald (inverse Gaussian) distribution.
weibull      Weibull distribution.
zipf         Zipf's distribution over ranked data.
=====

```

Chapter 7, Visual Finance via Matplotlib

```

C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

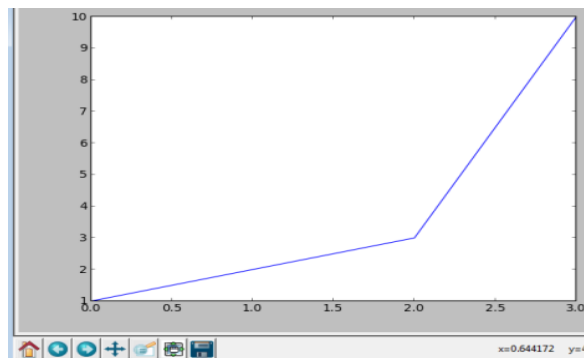
C:\Users\yany>cd\Python27
C:\Python27>

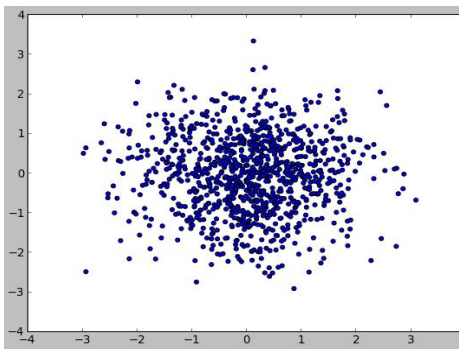
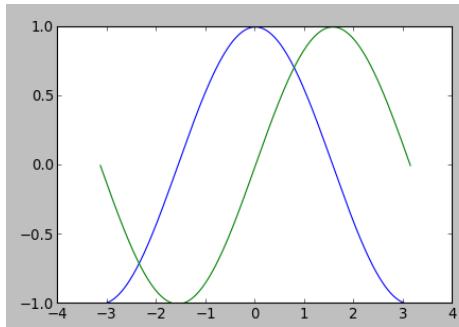
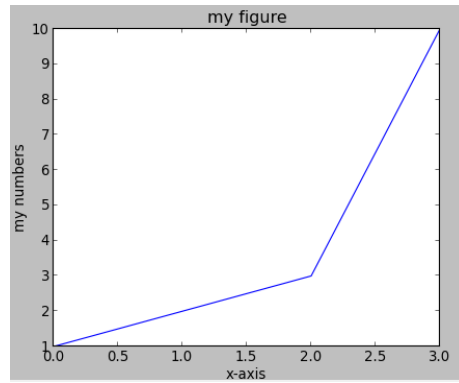
```

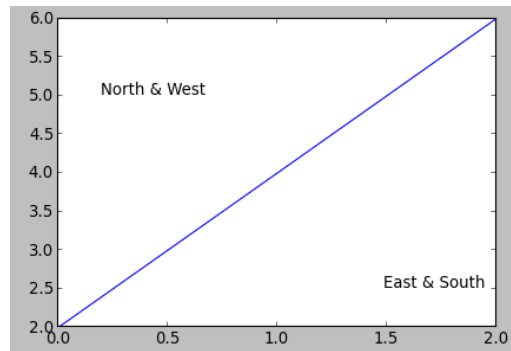
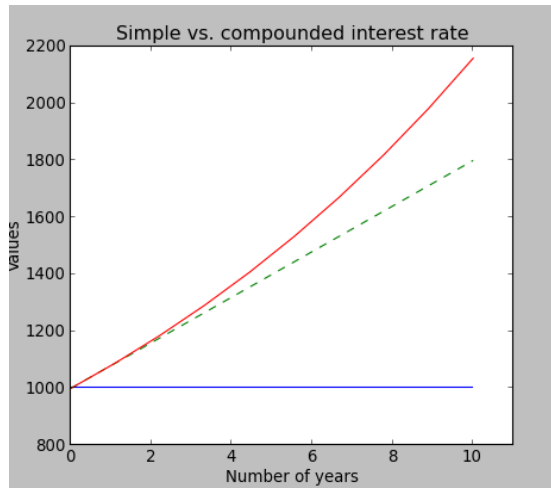
```
C:\Python27>pip install matplotlib
Get: [pypi-be.activestate.com] :repository-index:
Get: [pypi-free.activestate.com] :repository-index:
autosync: synced 2 repositories
Expanding "%APPDATA%\Python\Scripts" in user's %PATH%; please launch a new Command Window
The following packages will be installed into "%APPDATA%\Python" (2.7):
  as.mklruntime-1.3 numpy-1.7.1 matplotlib-1.1.1
Get: [pypi-free.activestate.com] as.mklruntime 1.3
Get: [pypi-free.activestate.com] matplotlib 1.1.1
Get: [pypi-free.activestate.com] numpy 1.7.1
Installing as.mklruntime-1.3
```

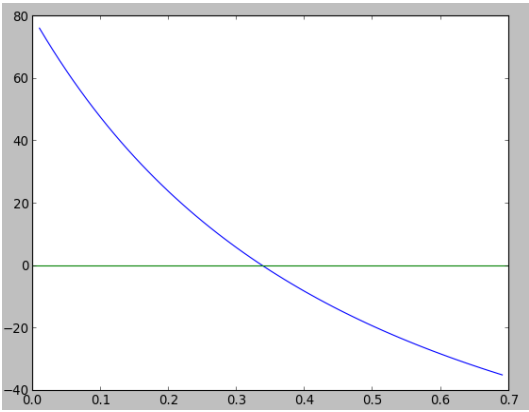
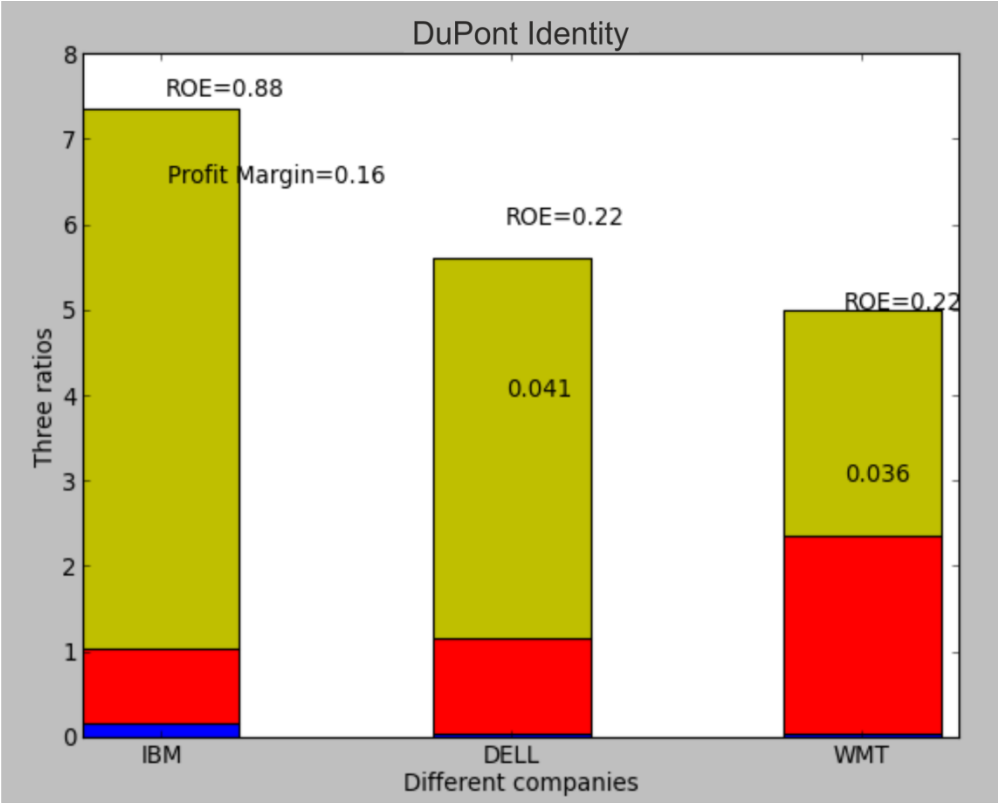
```
C:\Python27>pip install scipy
autosync: nothing new to sync
The following packages will be installed into "%APPDATA%\Python" (2.7):
  scipy-0.10.1
Get: [pypi-free.activestate.com] scipy 0.10.1
downloading: [=====] | 42% 7.4MB/17.7MB (9s; 13s left)
```

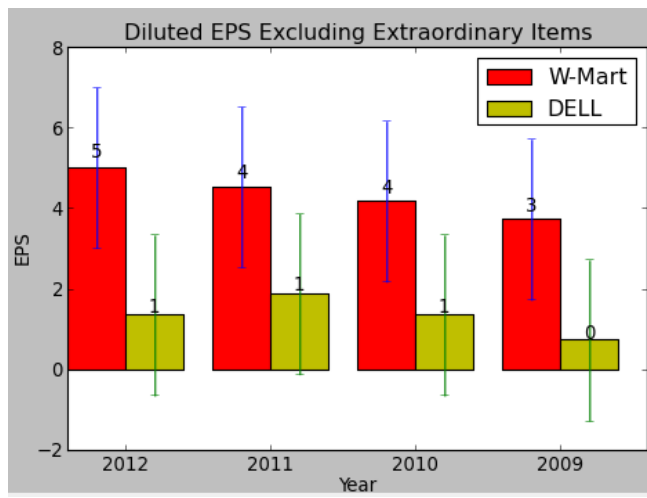
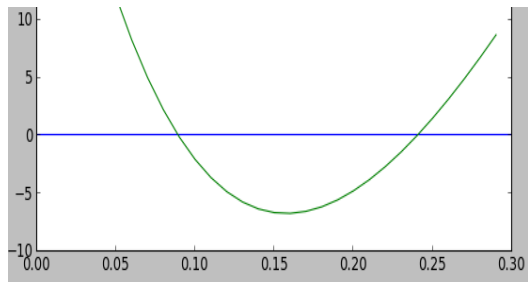
- ActiveState ActivePython 2.7 (32-bit)
- ActivePython Documentation
- IDLE (Python GUI)
- Python Interactive Shell
- Python Package Manager (PyPM)
- Uninstall ActivePython

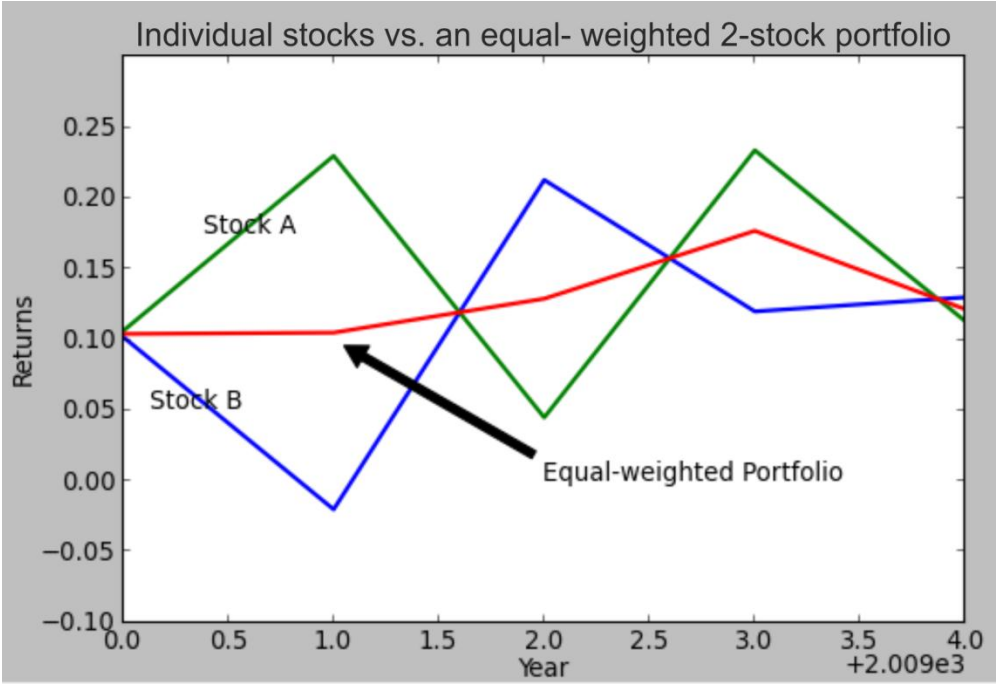


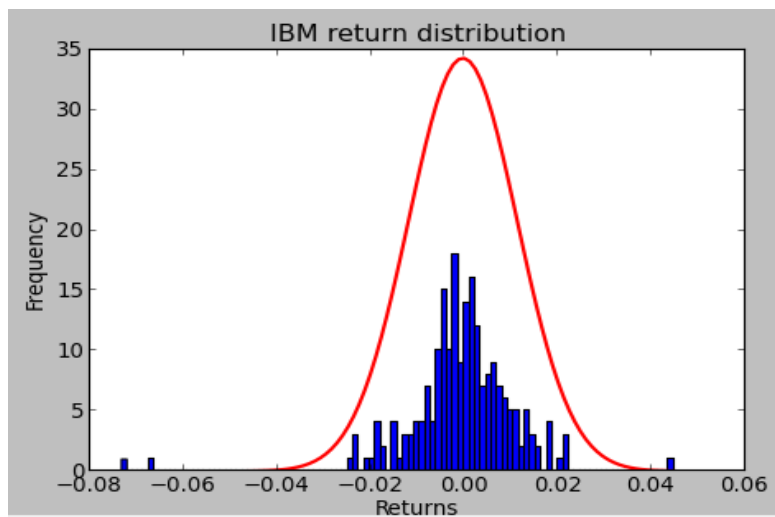
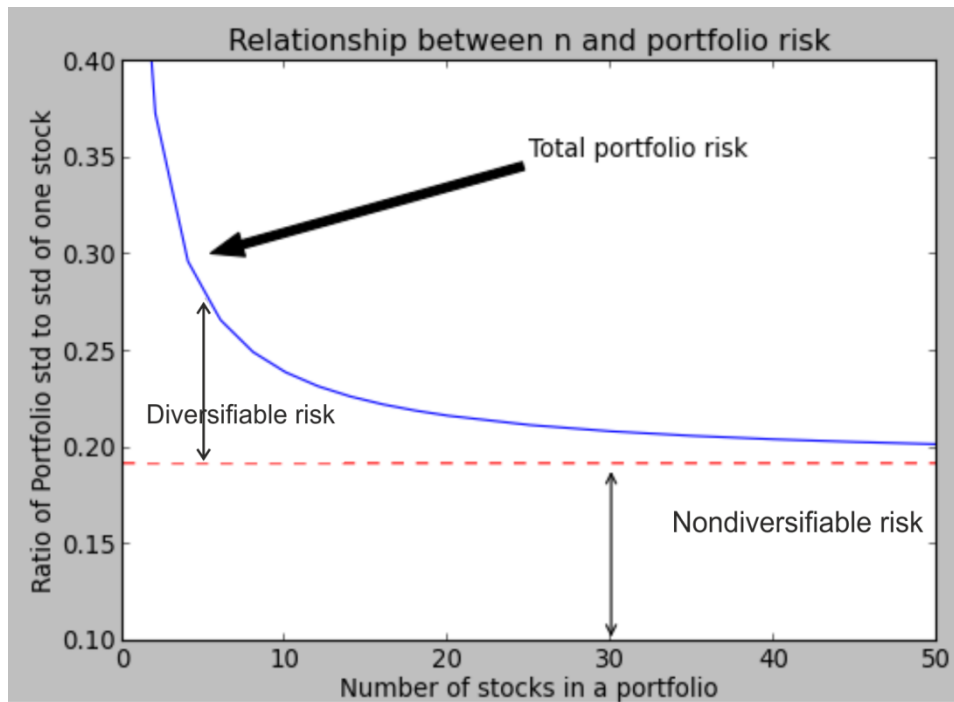


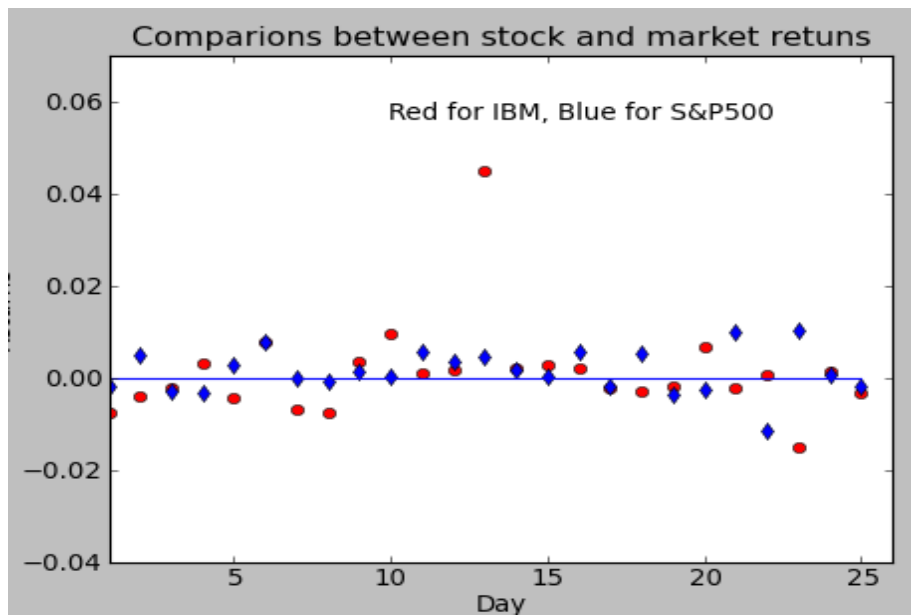
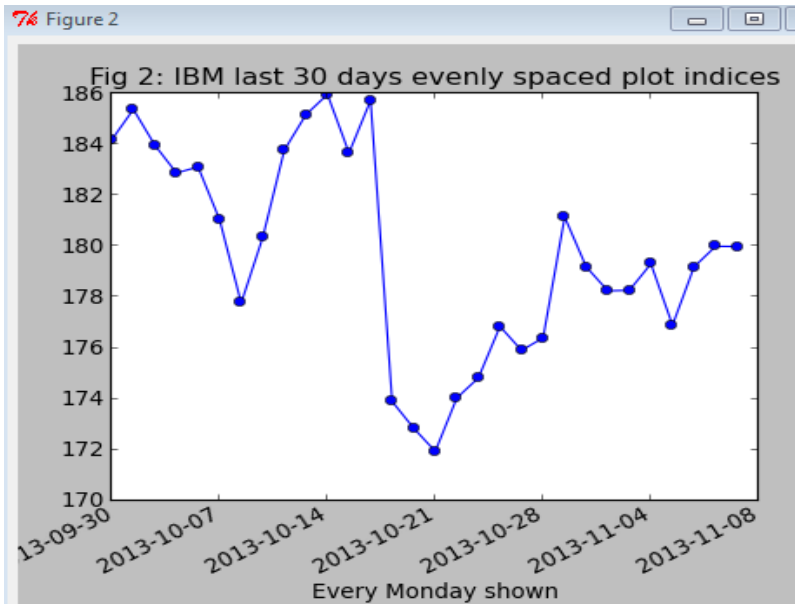




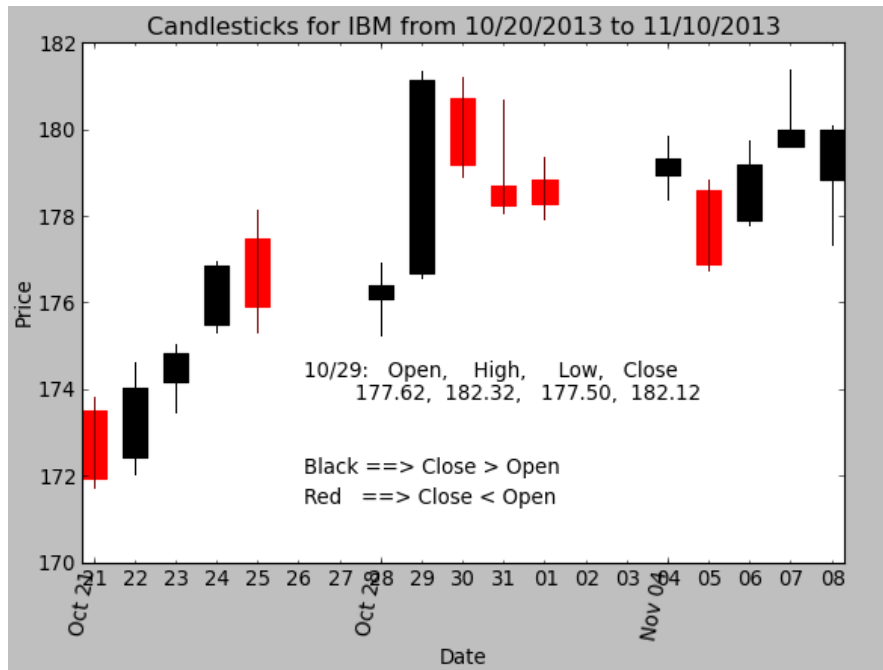
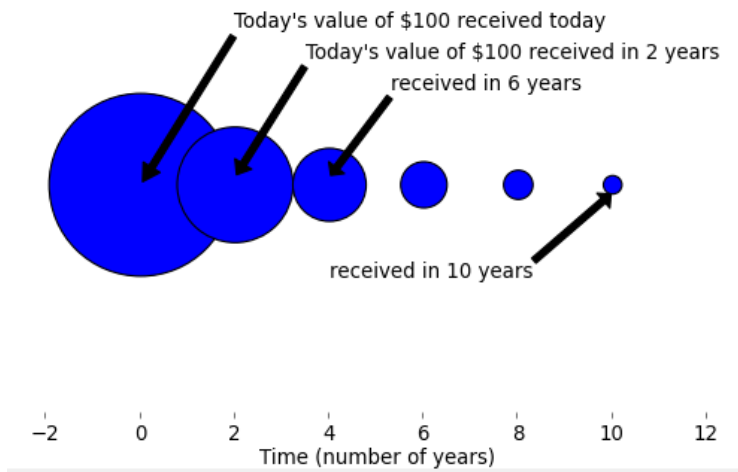


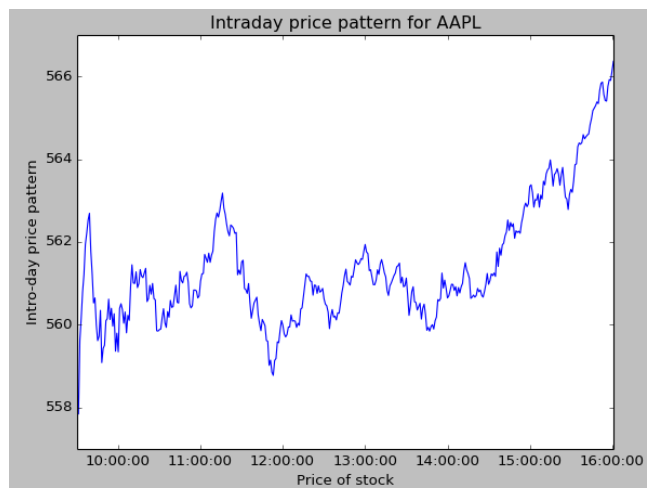
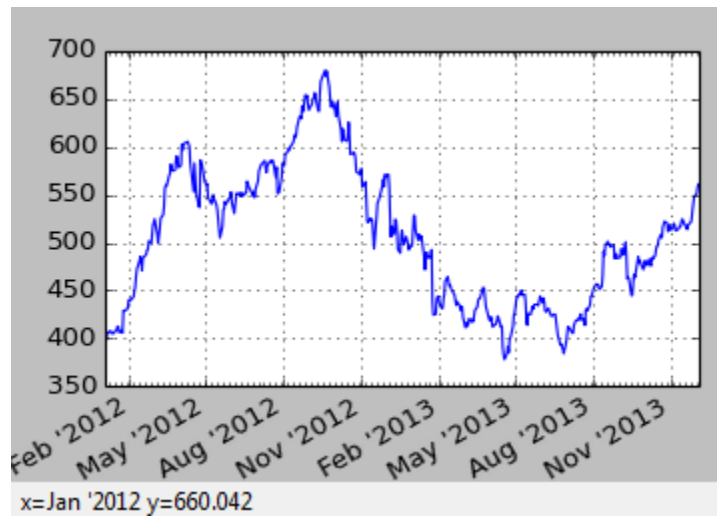




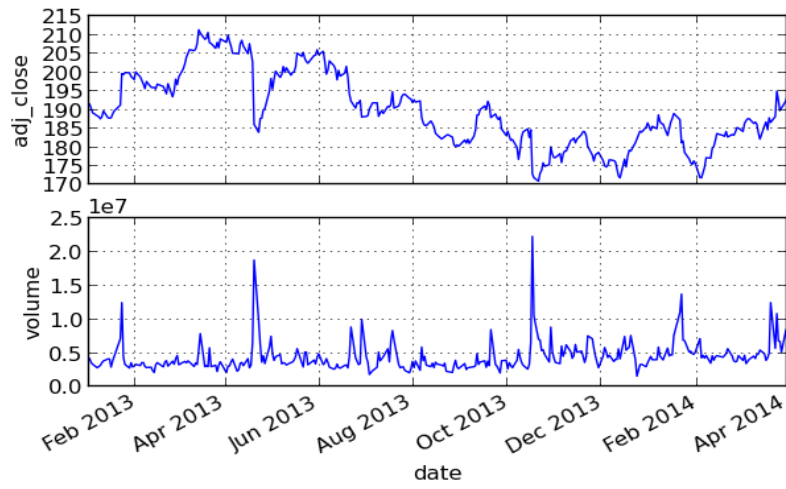
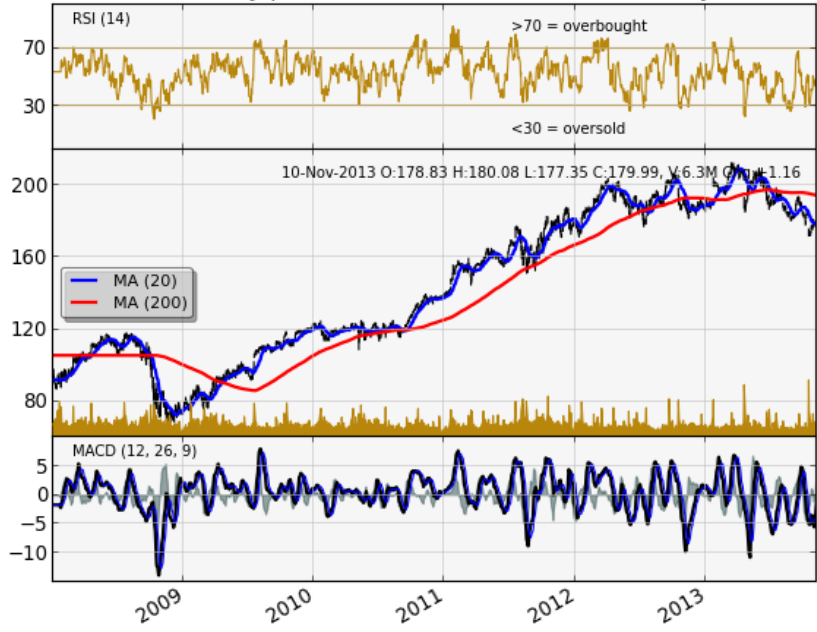


Time value of money





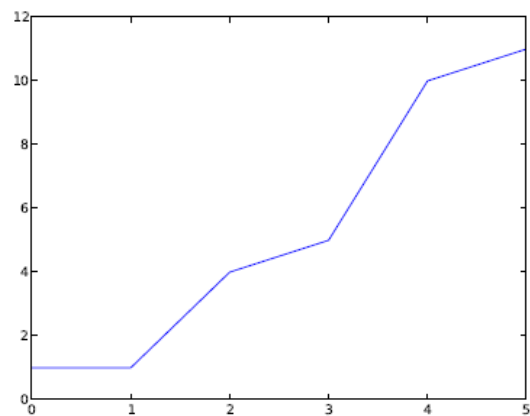
IBM daily price moment from 1/1/2009 to today

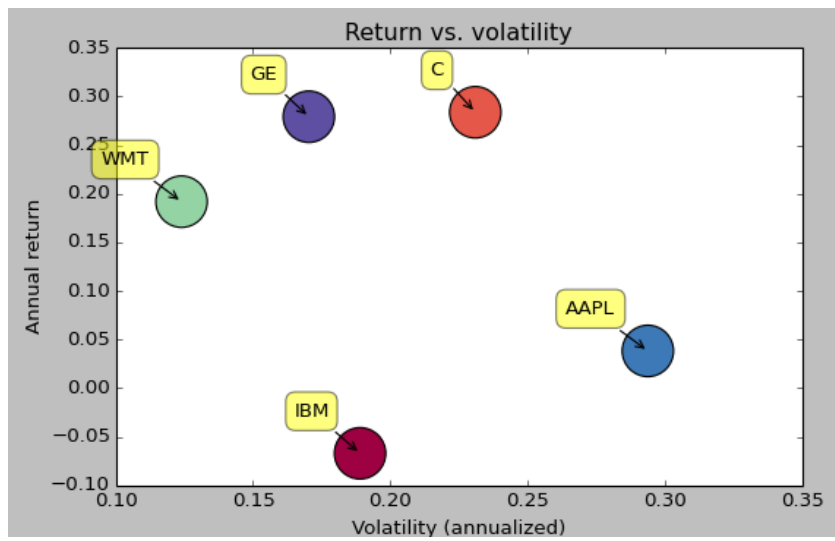
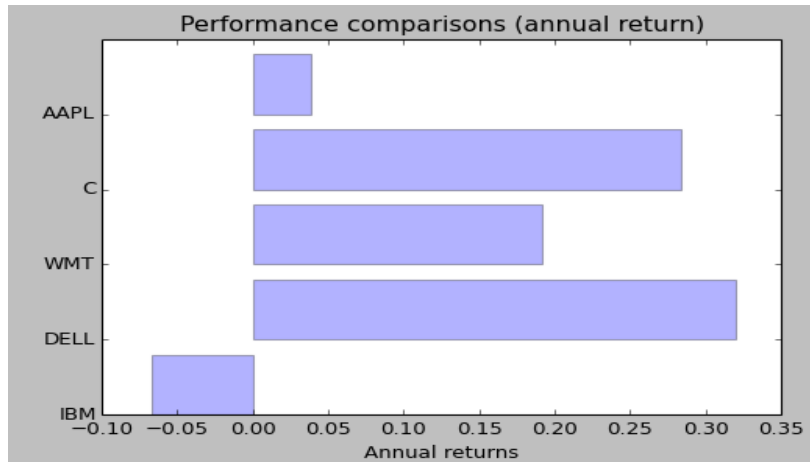


$$c = S_0 N(d_1) - Ke^{-rT} N(d_2)$$

$$d_1 = \frac{\ln(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

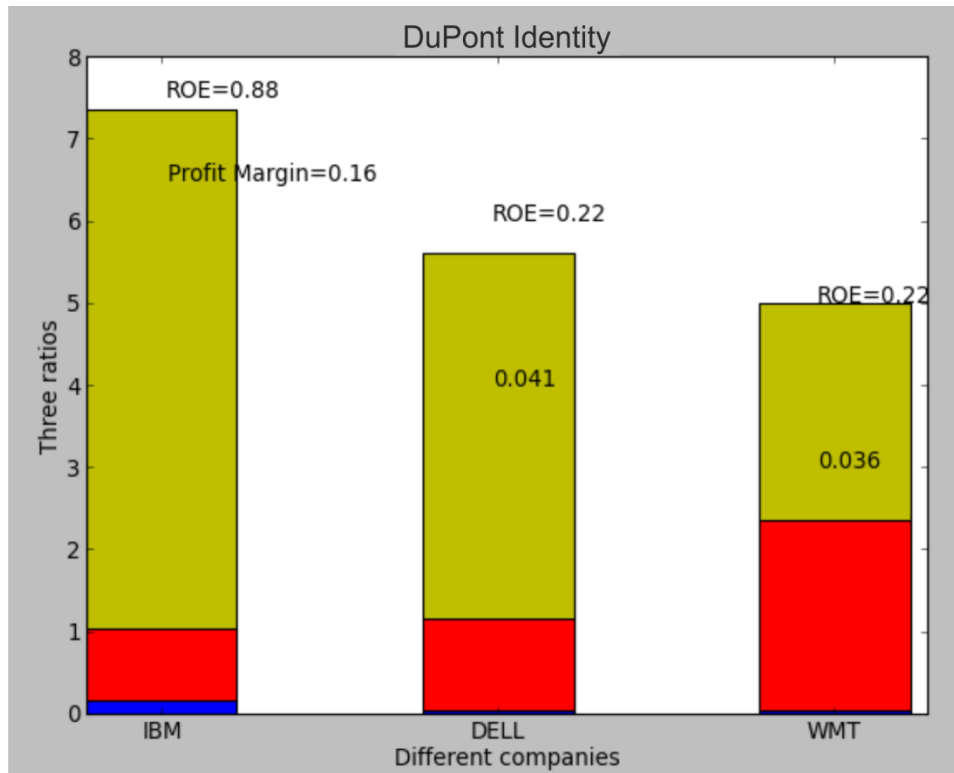
$$d_2 = \frac{\ln(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$



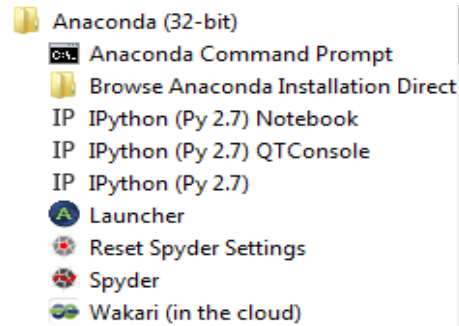


Recent	% \$
AAPL	+0.51%
DELL	0.00%
IBM	+0.19%
IBM.L	0.00%
WMT	-0.97%
^GSPC	-0.43%
^DJI	-0.43%
R	-0.51%

A		
^NSEI	CNX NIFTY	Index-NSE
^DJI	Dow Jones Industrial Average	Index-DJI
^GSPC	S&P 500	Index-SNP
^IXIC	NASDAQ Composite	Index-NIM
^VIX	VOLATILITY S&P 500	Index-Chicago Board.
^JKSE	Composite Index	Index-Jakarta
^NDX	NASDAQ-100	Index-NIM
^N225	Nikkei 225	Index-Osaka Stock Ex.
^TNX	CBOE Interest Rate 10-Year T-No	Index-Chicago Board.
^HSI	HANG SENG INDEX	Index-Hong Kong



Chapter 8, Statistical Analysis of Time Series



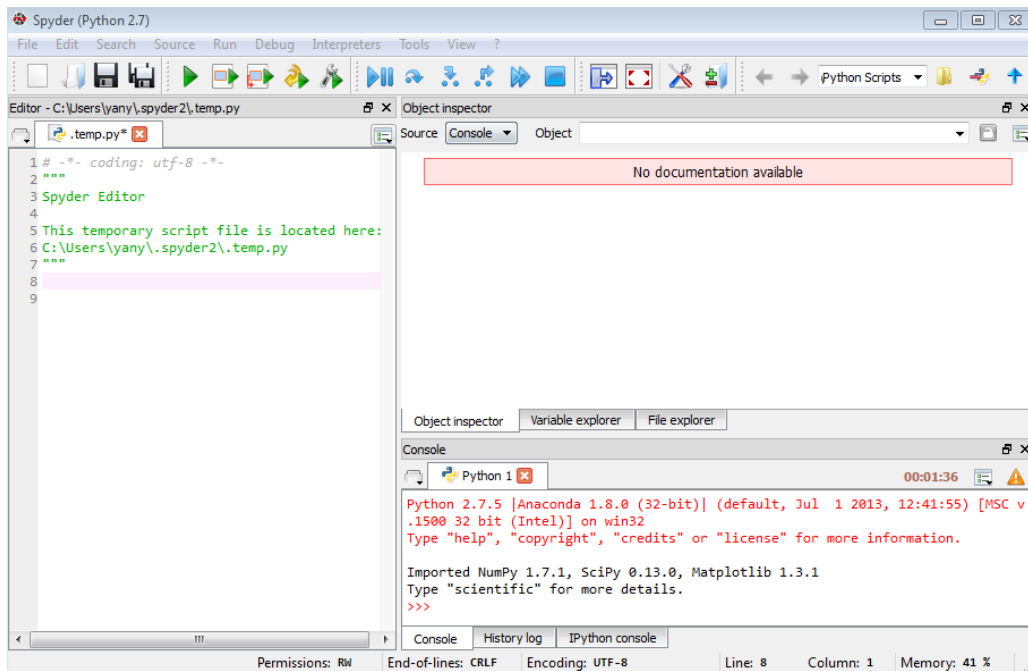
```
C:\Users\yany\AppData\Local\Continuum\Anaconda>python
Python 2.7.5 |Anaconda 1.8.0 (32-bit)| (default, Jul 1 2013, 12:41:55) [MSC v
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas as pd
>>> import statsmodels as sm
>>>
```

```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\yany>cd\temp

C:\temp>type test01.py
x=10
print x
C:\temp>python test01.py
10

C:\temp>
```



```
>>> runfile('C:/temp/test01.py', wdir=r'C:/temp')
10
>>>
```

```
>>> p[0:3]
rec.array([(datetime.date(2013, 1, 2), 2013, 1, 2, 734870.0, 10.146067415730338, 10.5, 10.5196
62921348315, 10.126404494382024, 26421700.0, 10.5),
          (datetime.date(2013, 1, 3), 2013, 1, 3, 734871.0, 10.435557586837295, 10.75, 11.13322669
1042047, 10.406078610603291, 38131300.0, 10.75),
          (datetime.date(2013, 1, 4), 2013, 1, 4, 734872.0, 10.740692798541476, 10.78, 10.87826800
3646307, 10.622771194165907, 18706400.0, 10.78)],
          dtype=[('date', '<O>'), ('year', '<i2>'), ('month', '<i1>'), ('day', '<i1>'), ('d', '<f8>'), ('op
en', '<f8>'), ('close', '<f8>'), ('high', '<f8>'), ('low', '<f8>'), ('volume', '<f8>'), ('aclose', '<
f8>')])
>>>
```

	A	B	C
1	1/1/2013	0.1	0.3
2	1/2/2013	0.2	0.4

```

>>>
>>> yyyymm[0:5]
['201301', '201301', '201301', '201301', '201301']
>>>
>>> .

```

```

>>> final.head()
      IBM_adjClose  Mkt_Rf      SMB      HML  Rf
20131001      184.37  0.0091  0.0039 -0.0013  0
20131002      182.97 -0.0010 -0.0036  0.0004  0
20131003      181.88 -0.0087 -0.0017  0.0020  0
20131004      182.12  0.0074  0.0000  0.0010  0
20131007      180.05 -0.0095 -0.0026  0.0006  0
>>> final.tail()
      IBM_adjClose  Mkt_Rf      SMB      HML  Rf
20131025      174.95  0.0033 -0.0035  0.0022  0
20131028      175.44  0.0009 -0.0005 -0.0002  0
20131029      180.16  0.0054 -0.0016 -0.0015  0
20131030      178.21 -0.0061 -0.0072  0.0040  0
20131031      177.28 -0.0034 -0.0008 -0.0036  0
>>> .

```

```

>>> print("      Today,           Price      High  Low, % from low ")
      Today,           Price      High  Low, % from low
>>> print(x[0], x[-1], high, low, round((x[-1]-low)/(high-low)*100,2))
(datetime.date(2014, 2, 28), 185.16999999999999, 211.42, 171.0, 35.06)
>>>

```

```

...
('Roll spread for ', 'IBM', 'is', 1.145)
>>>

```

```

const      2.699831e-03
x1         -1.297664e-13
x2         5.837434e-12
dtype: float64
>>>

```

OLS Regression Results

```

=====
Dep. Variable:          y      R-squared:          0.190
Model:                 OLS    Adj. R-squared:     0.187
Method:                Least Squares  F-statistic:       50.32
Date:                  Tue, 03 Dec 2013  Prob (F-statistic): 1.88e-11
Time:                  13:19:56  Log-Likelihood:    672.76
No. Observations:     216      AIC:               -1342.
Df Residuals:         214      BIC:               -1335.
Df Model:              1
=====

```

	coef	std err	t	P> t	[95.0% Conf. Int.]	
const	-0.0009	0.001	-1.237	0.217	-0.002	0.001
x1	0.7412	0.104	7.094	0.000	0.535	0.947

```

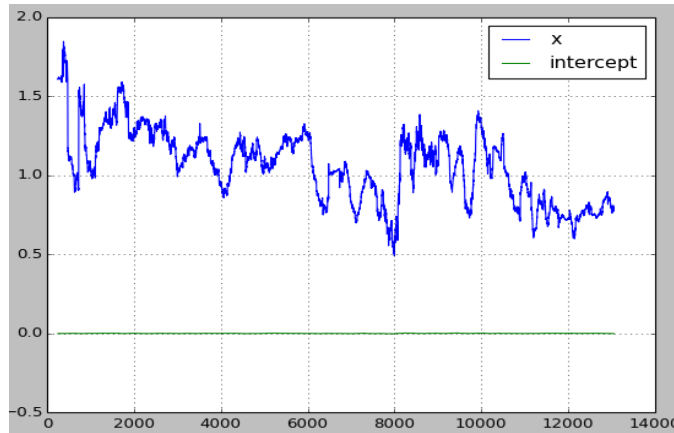
=====
Omnibus:                202.221  Durbin-Watson:      1.819
Prob(Omnibus):          0.000    Jarque-Bera (JB):   7283.884
Skew:                   -3.378    Prob(JB):           0.00
Kurtosis:               30.635    Cond. No.           142.
=====

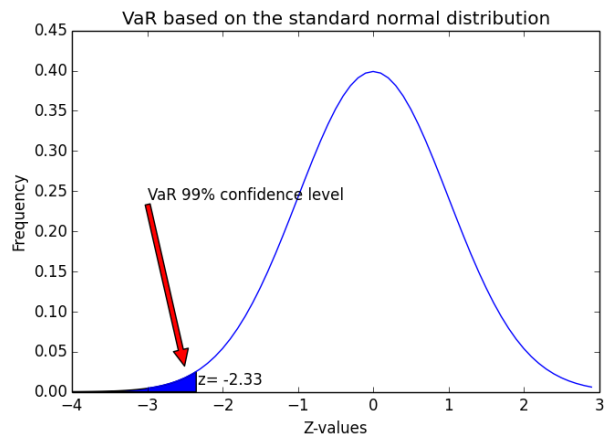
```

```

const    2.699831e-03
x1       -1.297664e-13
x2       5.837434e-12
dtype: float64
>>>

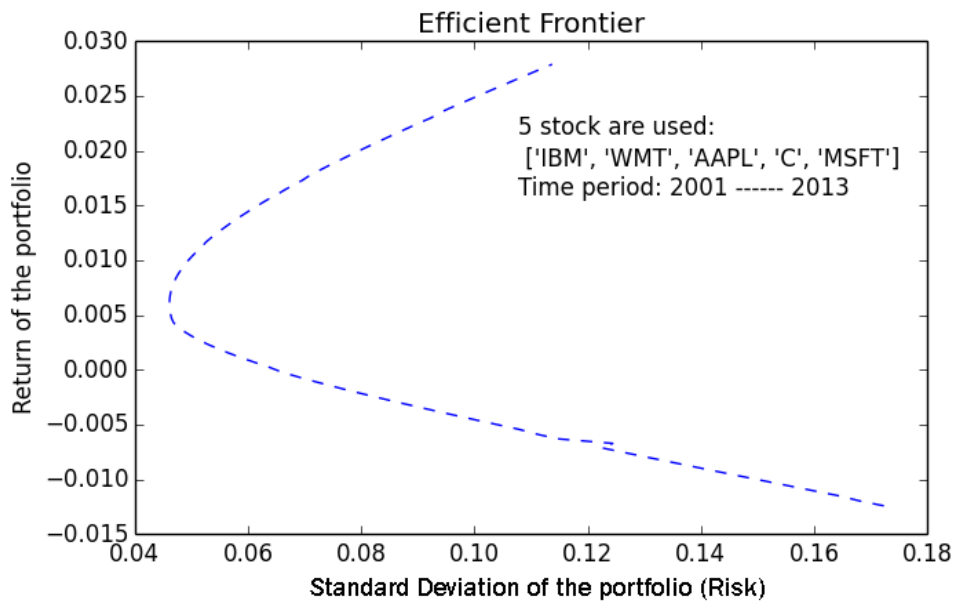
```



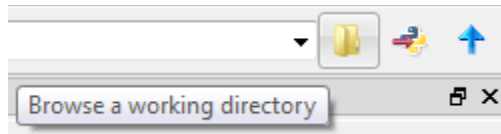
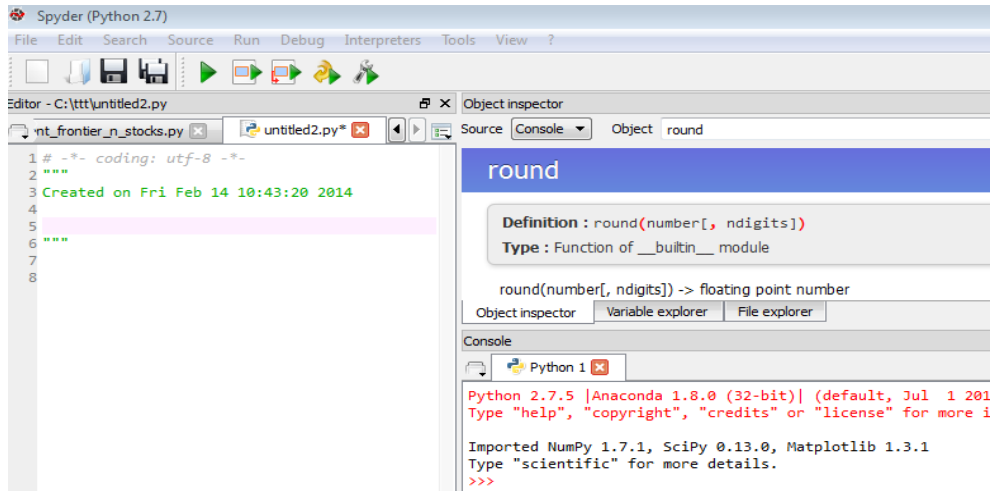


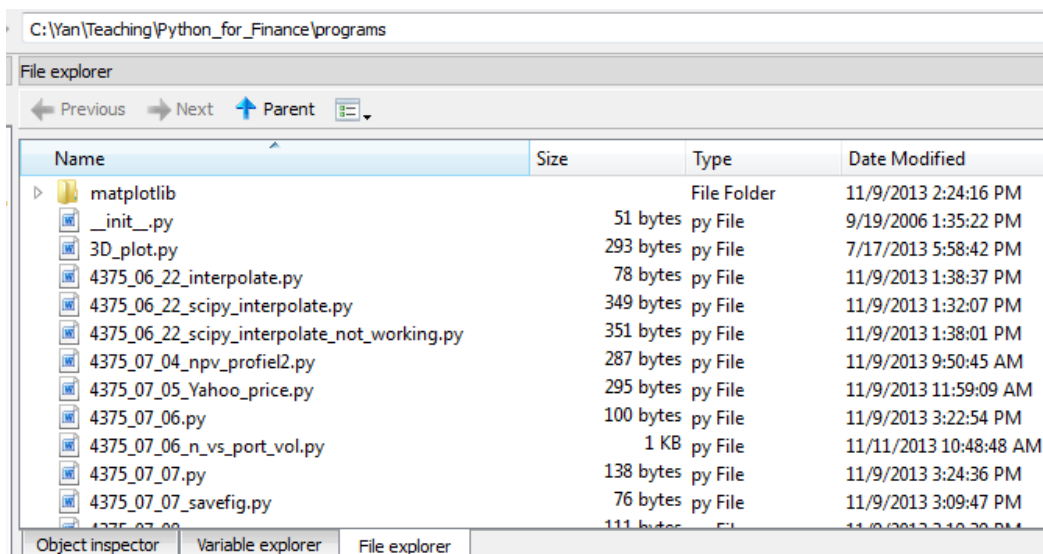
```
>>> print("return matrix", ret)
('return matrix', matrix([[ 0.1  ,  0.2  ],
 [ 0.1  ,  0.1071],
 [-0.02 ,  0.25  ],
 [ 0.012 ,  0.028 ],
 [ 0.06  ,  0.262 ],
 [ 0.14  ,  0.115 ]]))
>>> print(weight*covar_*weight.T)
[[ 0.10555915]]
>>>
```

```
Efficient portfolio (mean-variance) :ticker used
('IBM', 'WMT', 'C')
Sharpe ratio for an equal-weighted portfolio
[ 0.33333333  0.33333333  0.33333333]
0.634645504708
Optimization terminated successfully.
Current function value: -0.669702
Iterations: 30
Function evaluations: 58
Optimal weights are
[ 0.49713116  0.31047116  0.19239769]
final Sharpe ratio is
0.669701971388
>>>
```



```
>>> runfile('C:/k/43750S_08_49_high_freq_my11_best.py', wdir=r'C:/k')
AAPL 2014-02-25 09:30:00 [529.25 529.565 529.08 529.42 63319L]
AAPL 2014-02-26 09:30:00 [524.0 524.0 523.61 523.61 55958L]
AAPL 2014-02-27 09:30:00 [517.35 517.35 517.14 517.14 95704L]
AAPL 2014-02-28 09:30:00 [528.97 529.08 528.53 529.08 161320L]
AAPL 2014-03-03 09:30:00 [523.41 523.47 523.0 523.42 89110L]
>>>
```



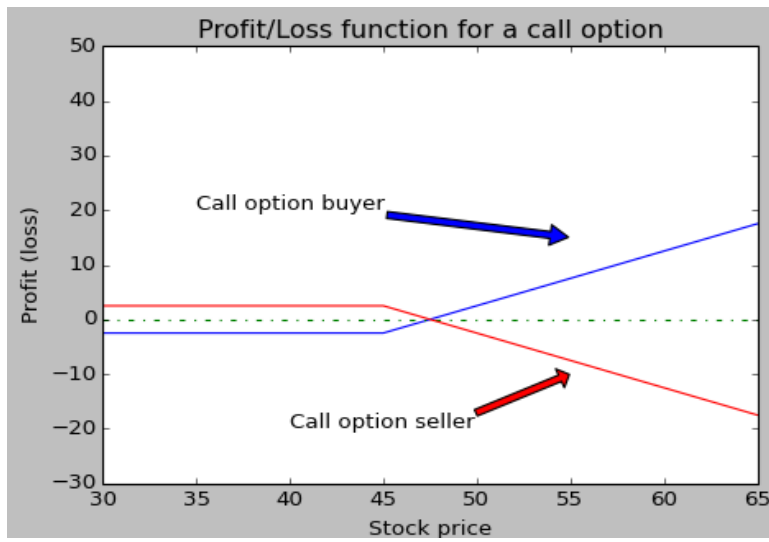
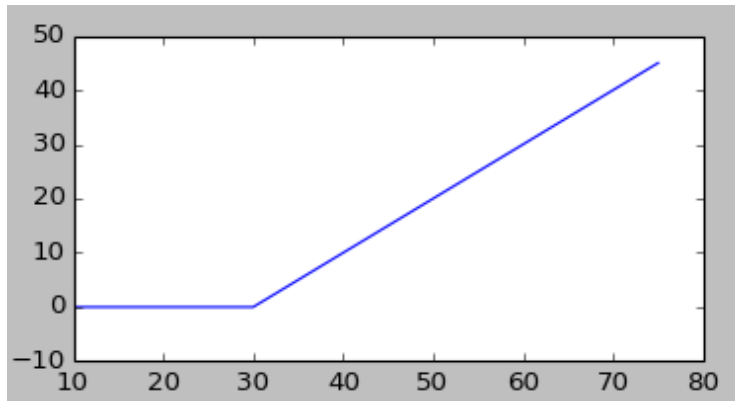


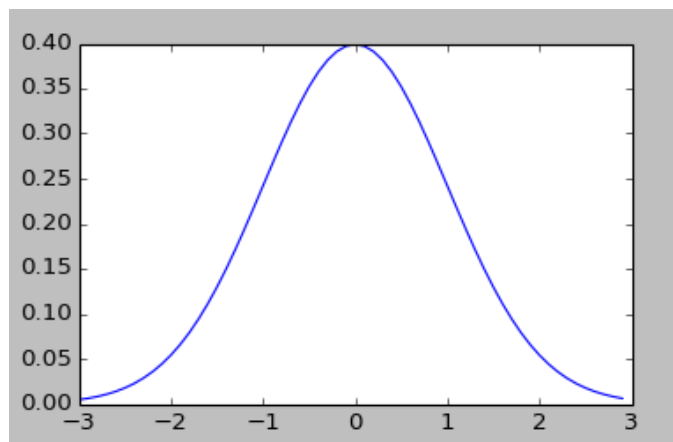
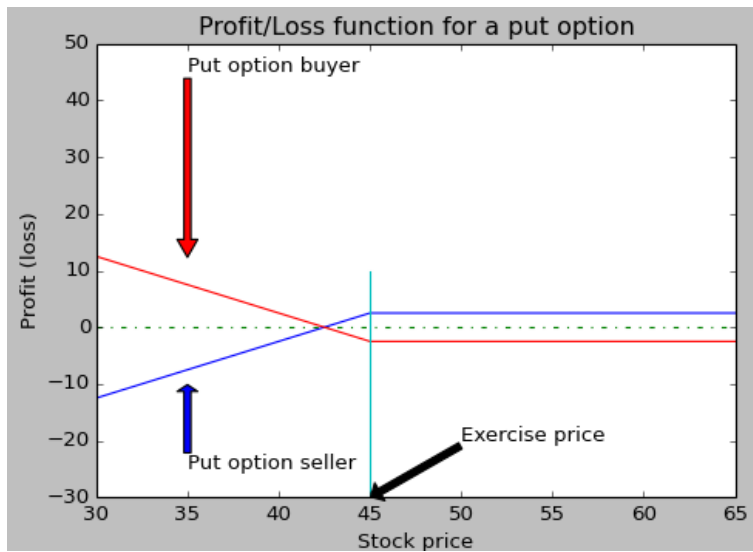
```

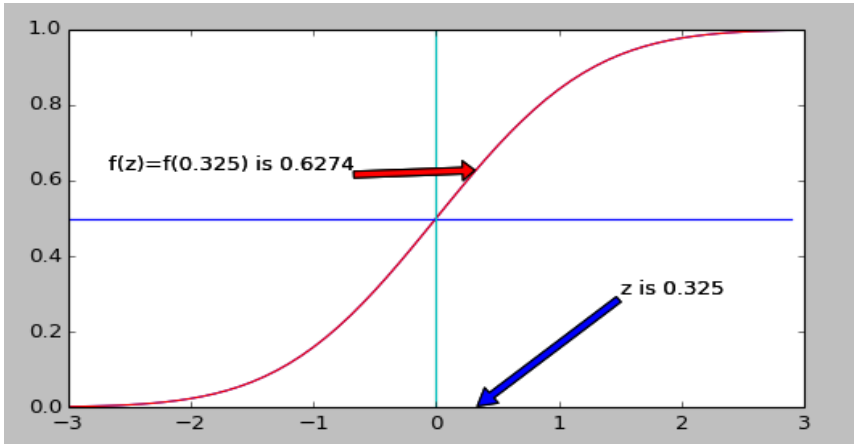
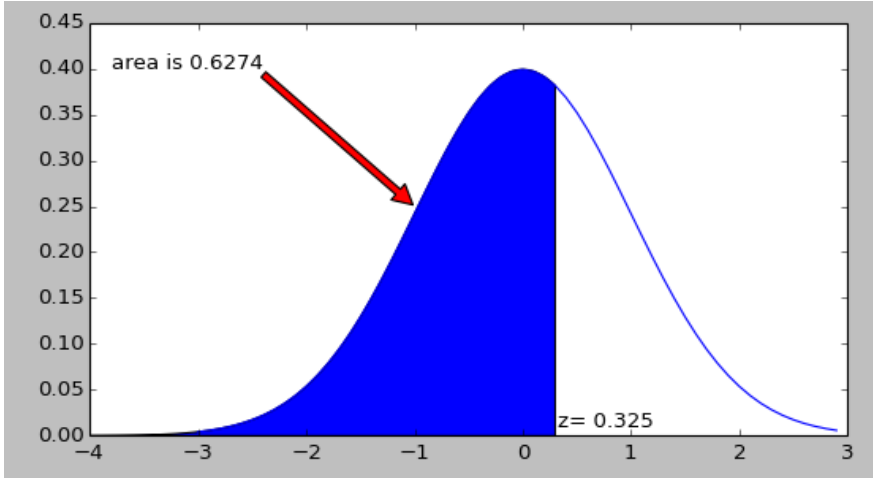
>>> t
array(['000001.SS', 'A', 'AA', 'AAPL', 'BC', 'BCF', 'C', 'CNC', 'COH',
      'CPI', 'DELL', 'GE', 'GOLDPRICE', 'GV', 'GVT', 'HI', 'HML', 'HPS',
      'HY', 'IBM', 'ID', 'IL', 'IN', 'INF', 'ING', 'INY', 'IO', 'ISL',
      'IT', 'J', 'JKD', 'JKE', 'JPC', 'KB', 'KCC', 'KFT', 'KIE', 'KO',
      'KOF', 'LBV', 'LCC', 'LCCM', 'LCM', 'LF', 'LG', 'LM', 'M', 'MA', 'MAA',
      'MD', 'MFL', 'MM', 'MPV', 'MY', 'Mkt_RF', 'NEV', 'NIO', 'NP', 'NU',
      'NYF', 'OI', 'OPK', 'PAF', 'PFO', 'PSJ', 'PZZA', 'Q', 'RH', 'RLV',
      'Rf', 'Russ3000E_D', 'Russ3000E_X', 'S', 'SBR', 'SCD', 'SEF', 'SI',
      'SKK', 'SMB', 'STC', 'T', 'TA', 'TBAC', 'TEN', 'TK', 'TLT', 'TOK',
      'TR', 'TZE', 'UHS', 'UIS', 'URZ', 'US_DEBT', 'US_GDP2009dollar',
      'US_GDP2013dollar', 'V', 'VC', 'VG', 'VGI', 'VO', 'VV', 'WG',
      'WIFI', 'WMT', 'WR', 'XLI', 'XON', 'Y', 'YANG', 'Z', '^AORD',
      '^BSESN', '^CCSI', '^CSE', '^FCHI', '^FTSE', '^GSPC', '^GSPTSE',
      '^HSI', '^IBEX', '^ISEQ', '^JKSE', '^KLSE', '^KS11', '^MXQ',
      '^NZ50', '^OMX', '^STI', '^STOXX50E', '^TWII'], dtype=object)
>>> len(t)
129
>>>

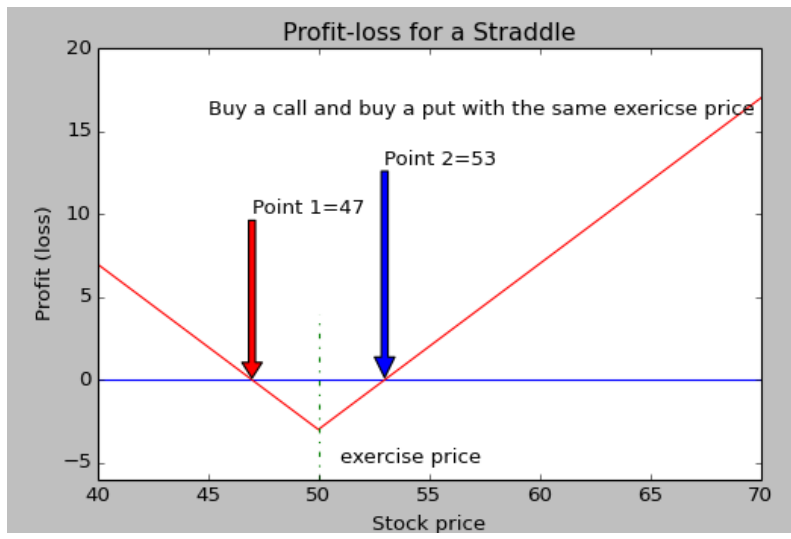
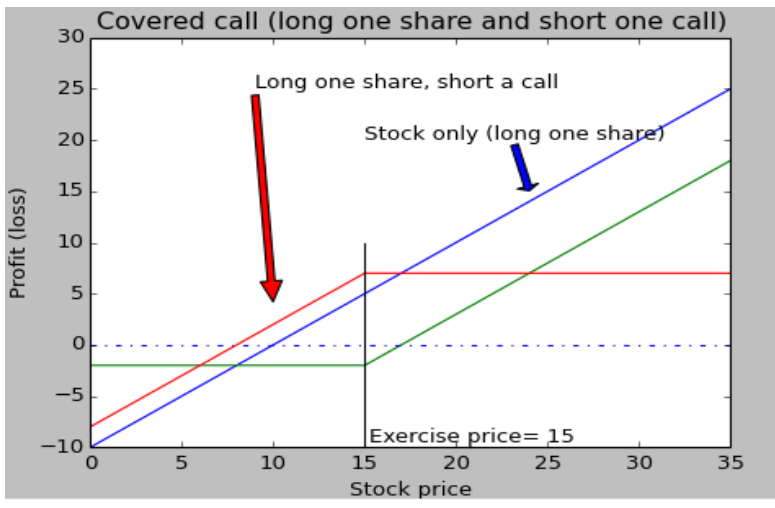
```

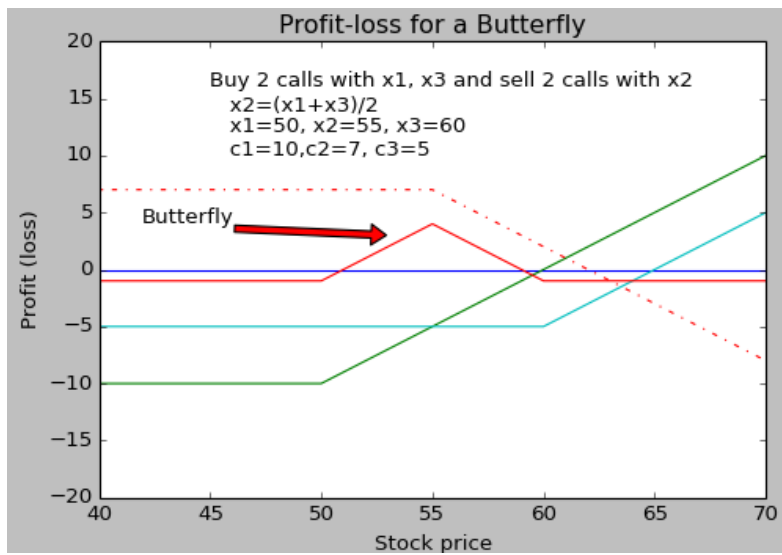
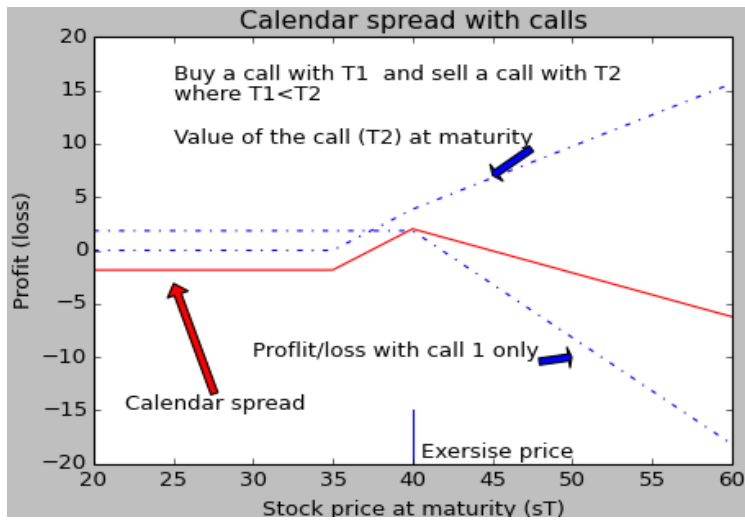
Chapter 9, The Black-Scholes-Merton Option Model

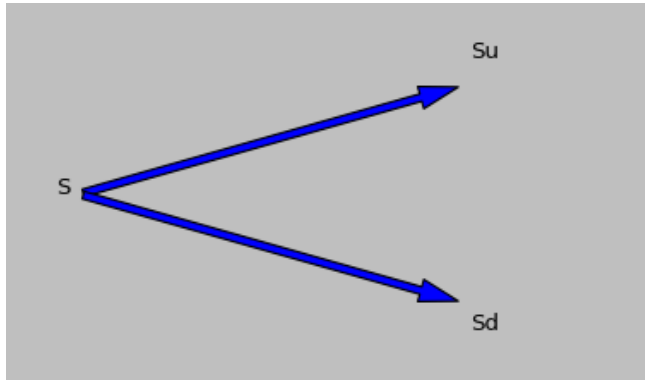
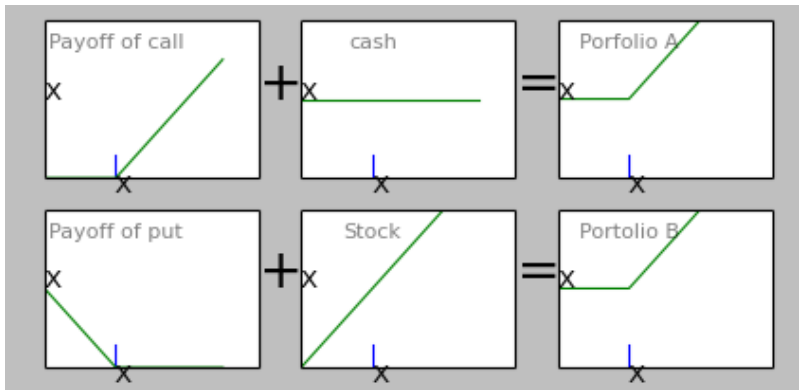


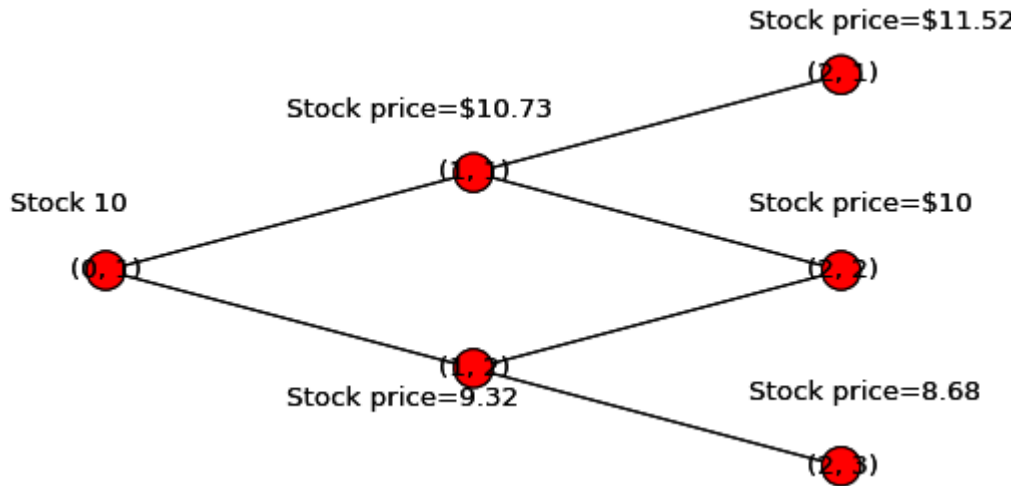




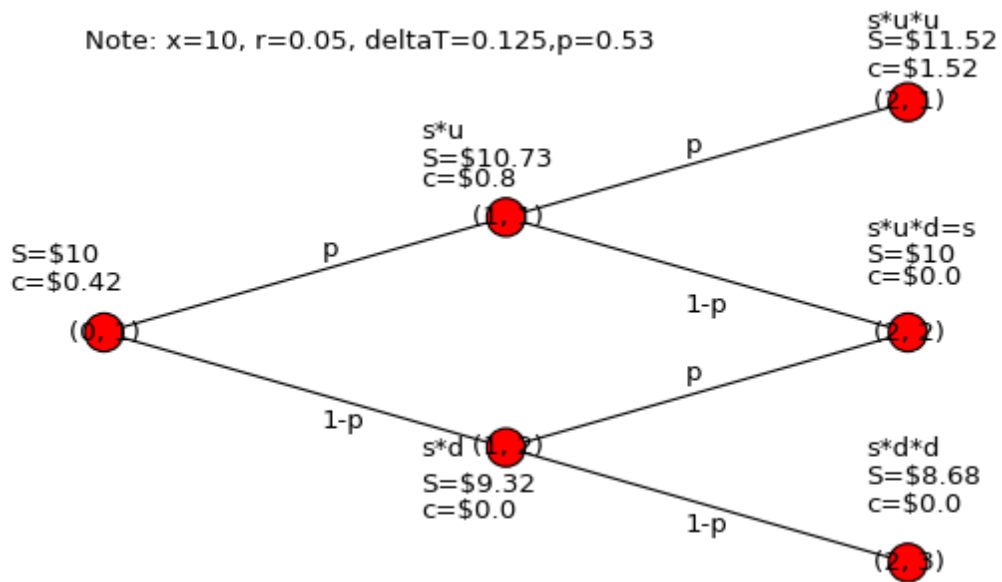


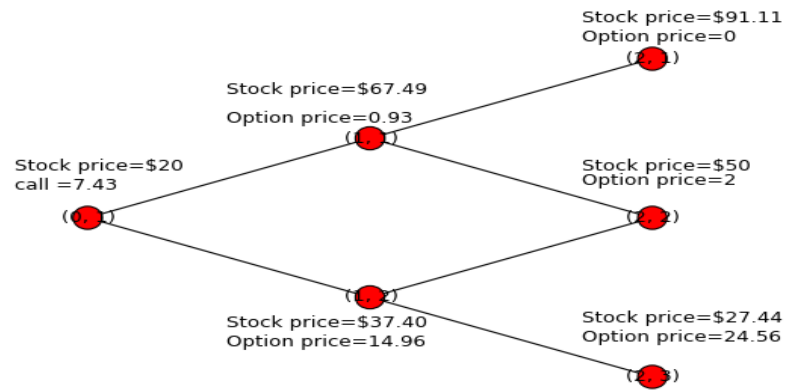




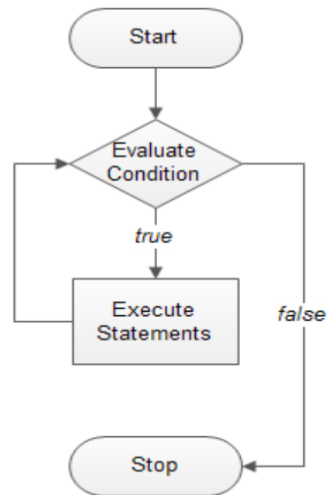


Note: $x=10$, $r=0.05$, $\Delta t=0.125$, $p=0.53$





Chapter 10, Python Loops and Implied Volatility



International Business Machines Corporation (IBM) - NYSE

[+ Add to Po](#)

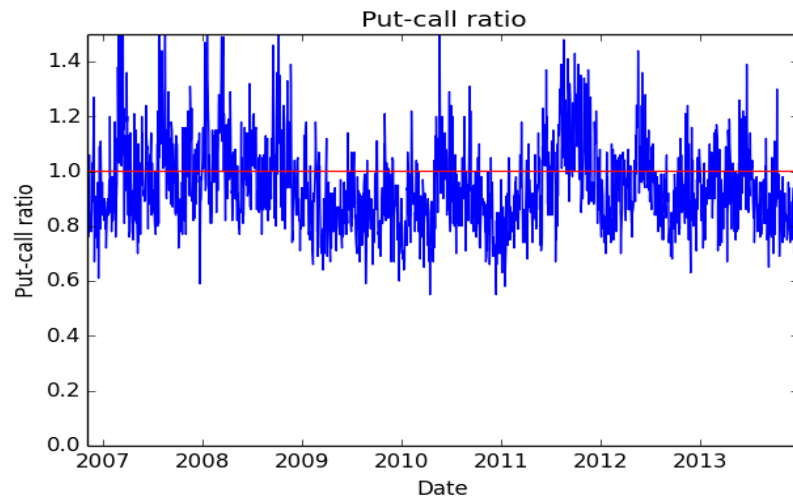
172.80 ↓ 0.57 (0.33%) Dec 13, 4:01PM EST | After Hours : **172.80** 0.00 (0.00%) Dec 13, 7:13PM EST

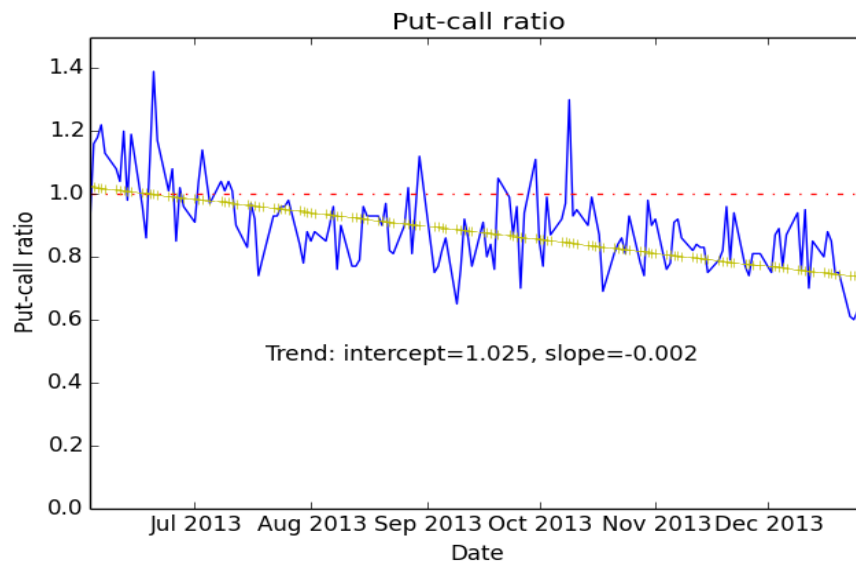
Options

Get Options for:

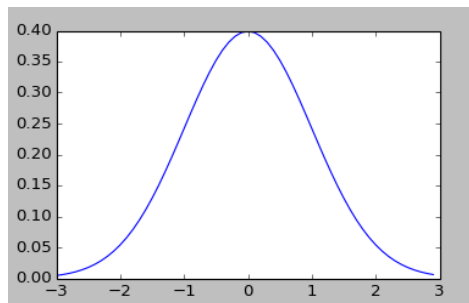
View By Expiration: [Dec 13](#) | [Jan 14](#) | [Feb 14](#) | [Apr 14](#) | [Jun 14](#) | [Jul 14](#) | [Jan 15](#) | [Jan 16](#)

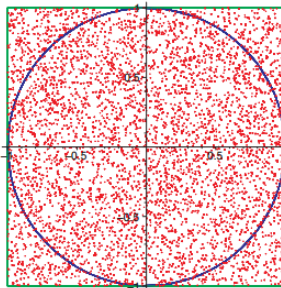
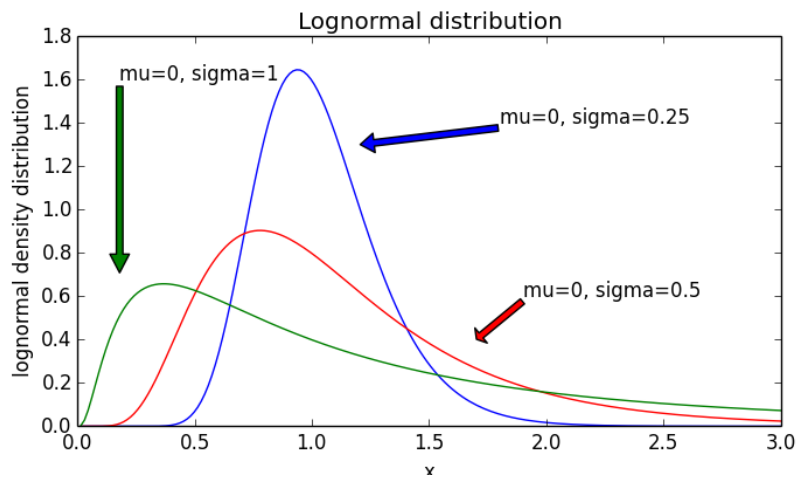
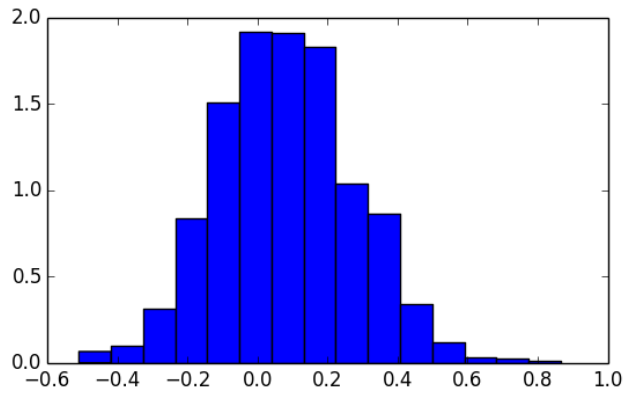
Call Options		Expire at close Friday, December 27, 2013					
Strike	Symbol	Last	Chg	Bid	Ask	Vol	Open Int
115.00	IBM131221C00115000	70.40	0.00	56.75	60.25	6	6
140.00	IBM131221C00140000	45.00	0.00	31.75	33.65	3	5
150.00	IBM131221C00150000	29.20	0.00	22.15	23.65	1	1





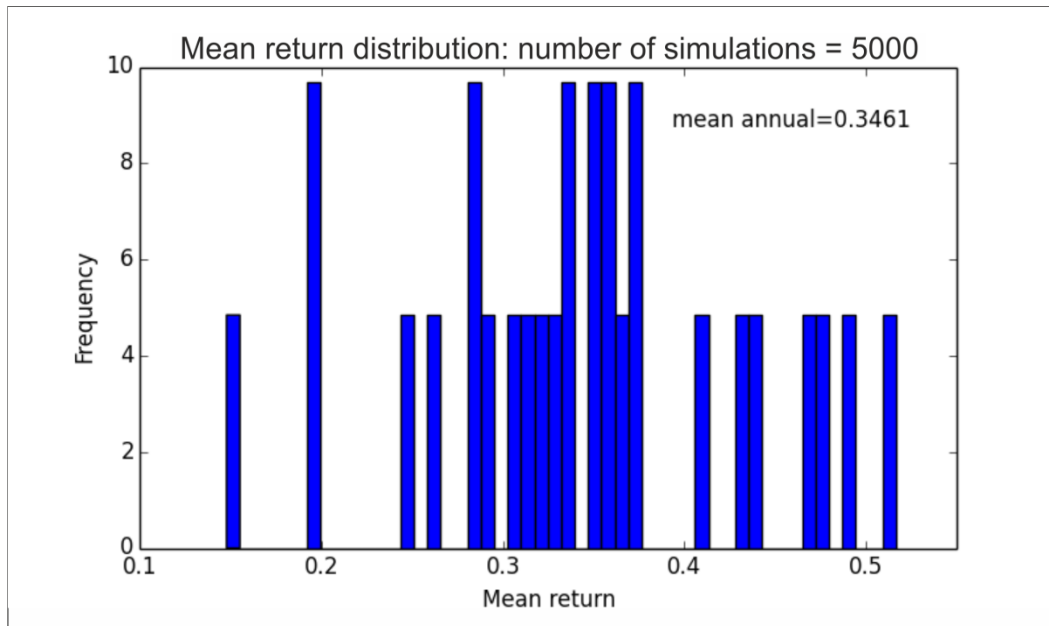
Chapter 11, Monte Carlo Simulation and Options

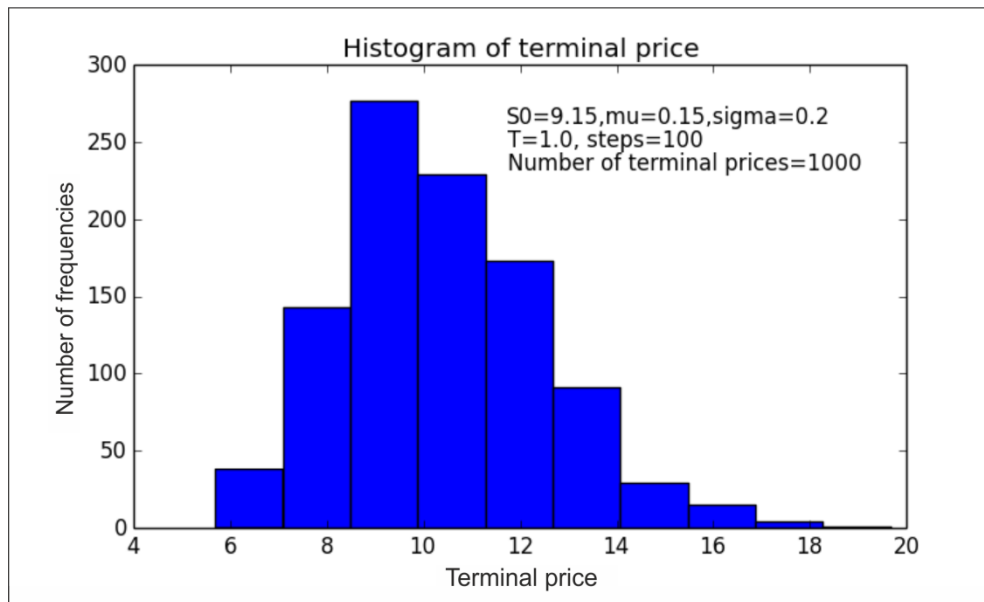
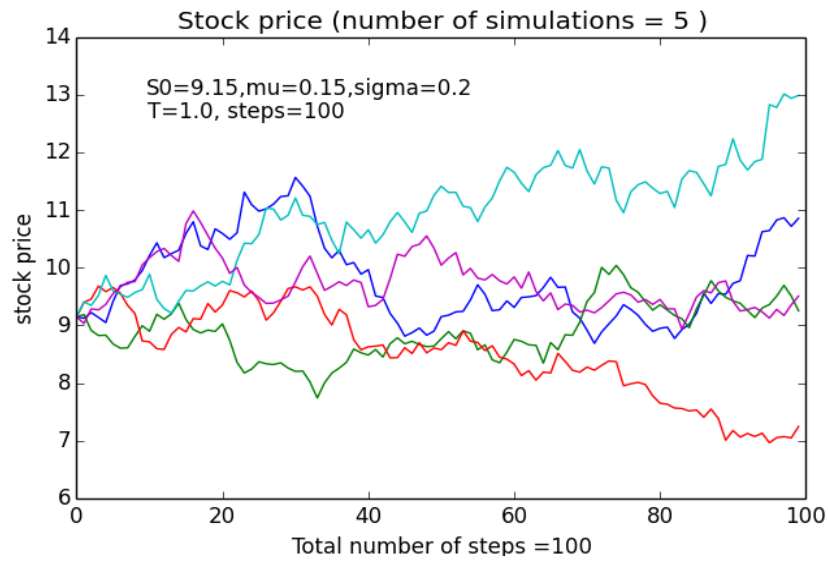


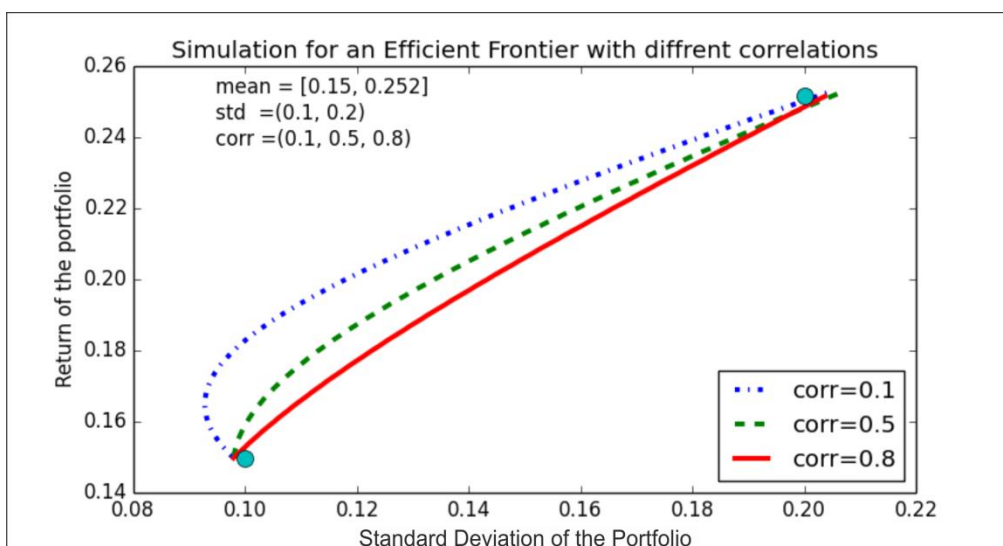
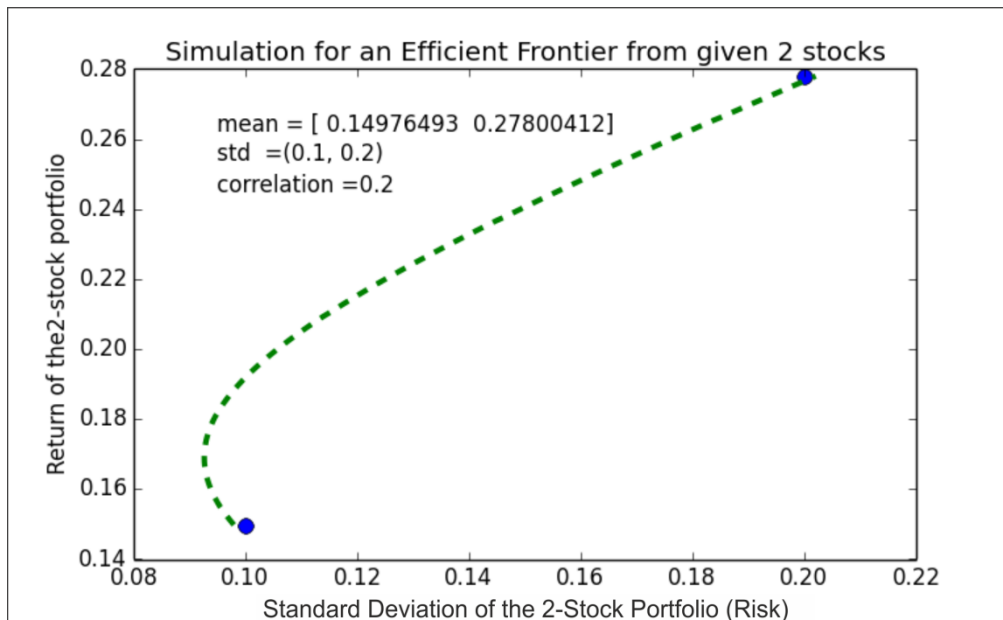


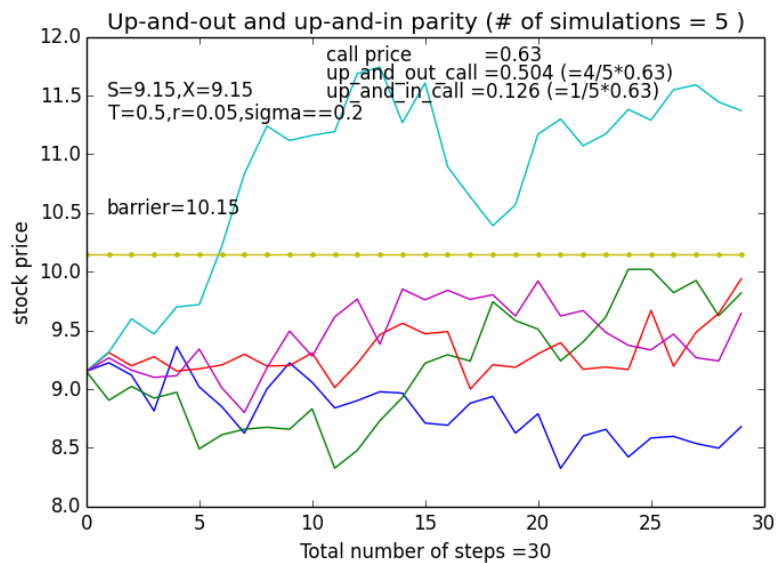
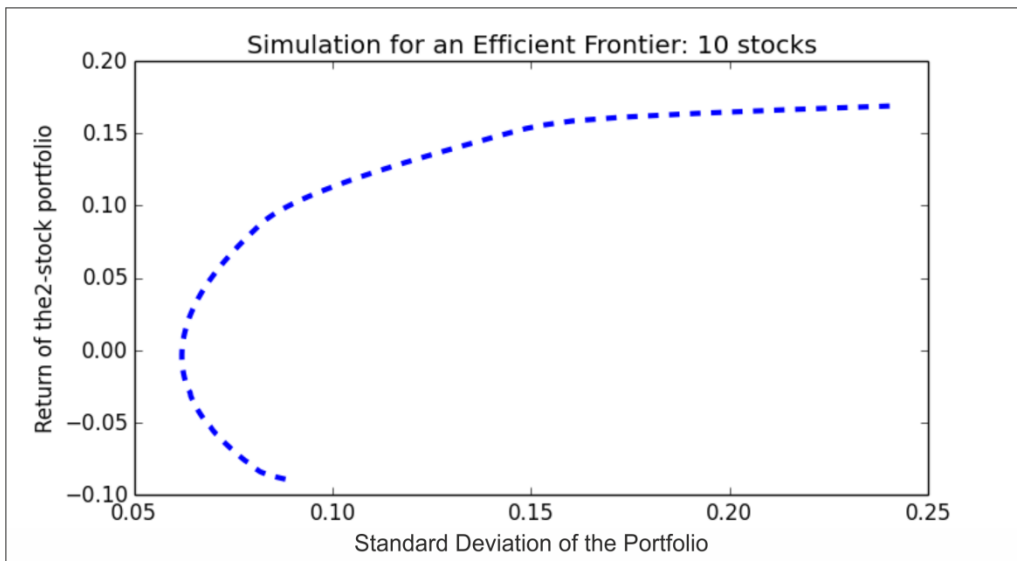
```
[ 1  2  4 10 17 20 21 24 31 70]
['IO', 'A', 'AA', 'KB', 'DELL', 'IN', 'INF', 'IBM', 'SKK', 'BC']
>>>
```

```
...
[ 7  6  9  1  2 10  5  8  3  4]
[ 8  2  4  6  1  9 10  3  7  5]
[ 4  9  1  6  8  2  5  7 10  3]
[ 2  1  6  9 10  5  7  4  8  3]
[ 7  3  9  5 10  4  8  6  2  1]
>>>
```









Chapter 12, Volatility Measures and GARCH


```

>>> print( 'ticker-',ticker,stats.shapiro(ret))
ticker- IBM (0.9316935539245605, 1.6079492248361013e-23)
>>> print('ticker=',ticker,'W-test, and P-value')
ticker= IBM W-test, and P-value
>>> print(stats.shapiro(ret))
(0.9316935539245605, 1.6079492248361013e-23)
>>>

```

```

>>> from scipy import stats,random
>>> import numpy as np
>>> ret = random.normal(0,1,500000)
>>> print( 'mean    =', np.mean(ret))
mean    = -0.00143927422656
>>> print( 'std     =',np.std(ret))
std     = 0.999594342278
>>> print( 'skewness=',stats.skew(ret))
skewness= 0.00198825271944
>>> print( 'kurtosis=',stats.kurtosis(ret))
kurtosis= -0.0106677615011
>>>

```

```

>>> >>> print( 'S&P500 n      =',len(ret))
S&P500 n      = 16102
>>> print( 'S&P500 mean    =',round(np.mean(ret),8))
S&P500 mean    = 0.00024465
>>> print( 'S&P500 std     =',round(np.std(ret),8))
S&P500 std     = 0.00981226
>>> print( 'S&P500 skewness=',round(stats.skew(ret),8))
S&P500 skewness= -1.50009683
>>> print( 'S&P500 kurtosis=',round(stats.kurtosis(ret),8))
S&P500 kurtosis= 38.21956241
>>>

```

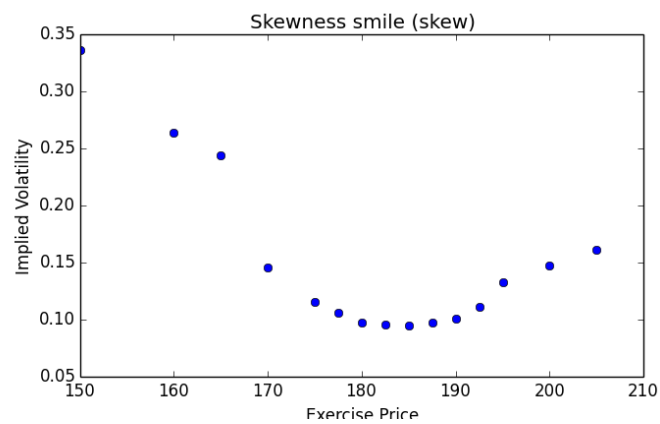
```

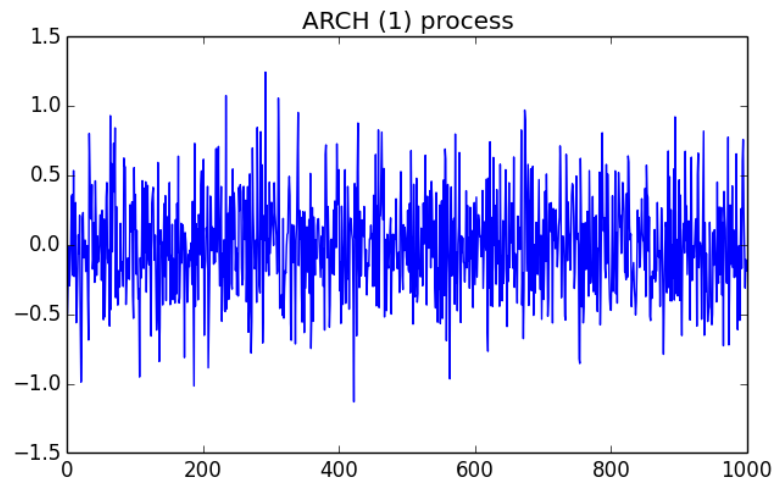
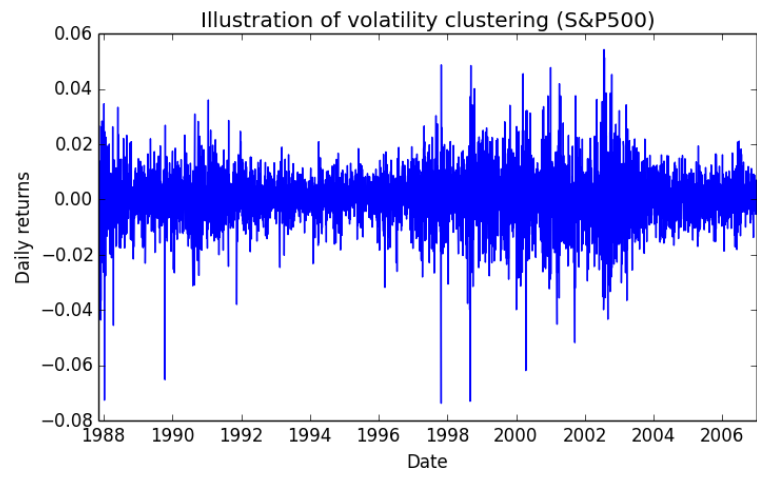
>>> print(' LPSD (annualized) for ', ticker, 'is ',round(LPSD,3))
LPSD (annualized) for IBM is nan
>>>

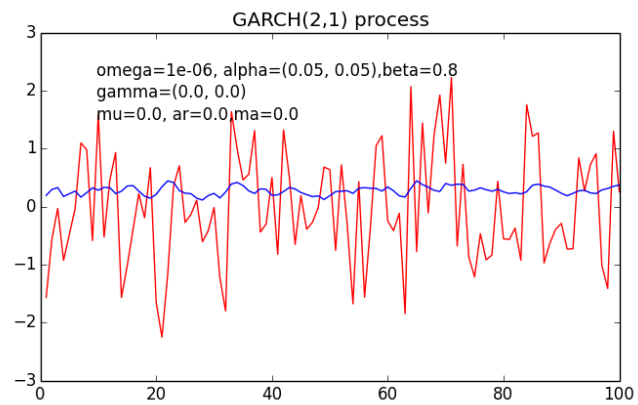
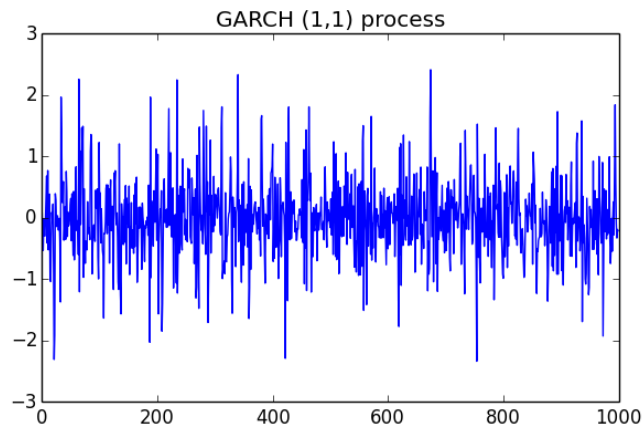
```

```
Std period #1 vs. std period #2
(0.020194, 0.017916)
T value , p-value
(18.059642154305671, 2.1409163403767813e-05)
>>>
```

```
y1 vs. x (we expect to accept the null hypothesis)
BP value, df, p-value
bp = (0.596446, 1.0, 0.5508776)
y2 vs. x (we expect to rject the null hypothesis)
BP value, df, p-value
('bp =', (17.611054, 1.0, 0.0))
>>>
```







```
Optimization terminated successfully. (Exit mode 0)
Current function value: -54.0664734199
Iterations: 12
Function evaluations: 94
Gradient evaluations: 12
```

>>>