Tsay_FinTS_ch2

April 9, 2018

In [44]:
```
"""debug example"""
# for i in [1,2,3,4,5]:
#     print(i)
    #import ipdb; ipdb.set_trace() # debugging starts here
    # http://frid.github.io/blog/2014/06/05/python-ipdb-cheatsheet/
```

Out[44]:
```
'debug example'
```

In [ ]:
```
# Tsay : Analysis of Financial Time Series, 2nd Edition
```

In [5]:
```
import random

import matplotlib.pyplot as plt
import numpy as np
```

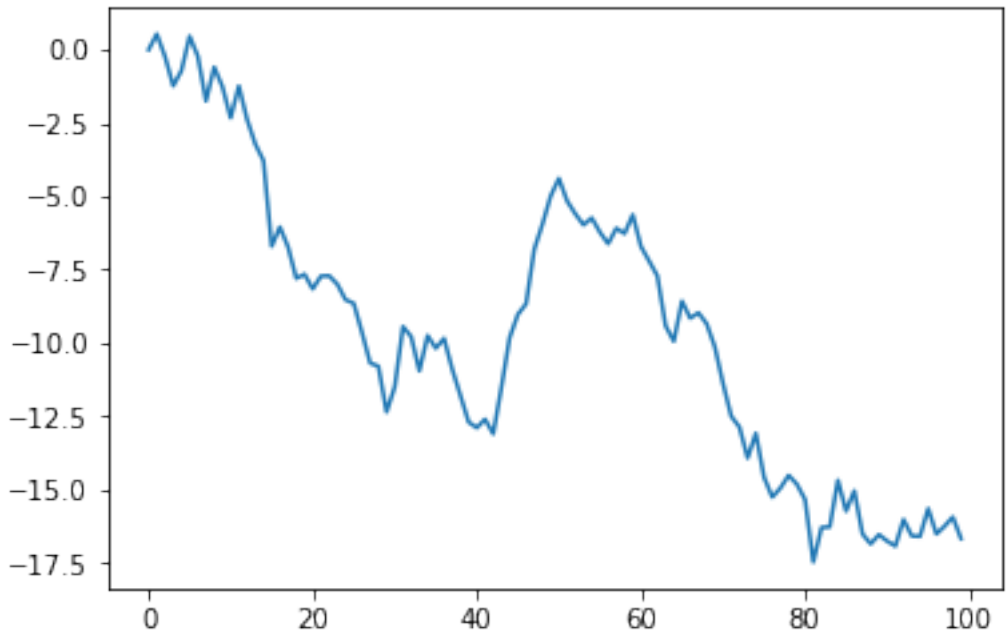
In [63]:
```
random.seed(9)
```

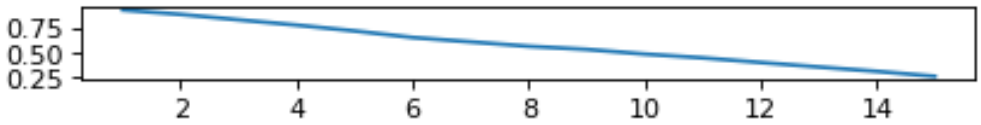
In [96]:
```
def white_noise(period = 100,distribution='uniform'):
    if distribution == 'uniform':
        return np.array([random.uniform(-1,1) for i in range(period)])

def random_walk(period = 100,initial=0):
    rw=[0]*period
    rw[0]=initial
    for i in range(1,100):
        rw[i]=rw[i-1]+random.gauss(0,1)
    return np.array(rw)



def plot(curve,period=100,step=1.0):
    plt.plot(np.array([i/step for i in range(period)]),curve)
    plt.show()

# autocorrelation function (ACF)
#vectorized version
def acf(series,lag=1):
    mean = np.mean(series)
```

```
        acf = np.sum(np.multiply(series[:-1-lag],series[lag:-1]))/np.sum(np.square(series))
        return acf

    def acf_curve(series,lag=15):
        plt.figure(num=None, figsize=(6, 0.5), dpi=80, facecolor='w', edgecolor='k')
        acf_curve = np.array([acf(series,i) for i in range(1,lag+1)])
        plt.plot(np.array(range(1,lag+1)),acf_curve)
        plt.show()
```

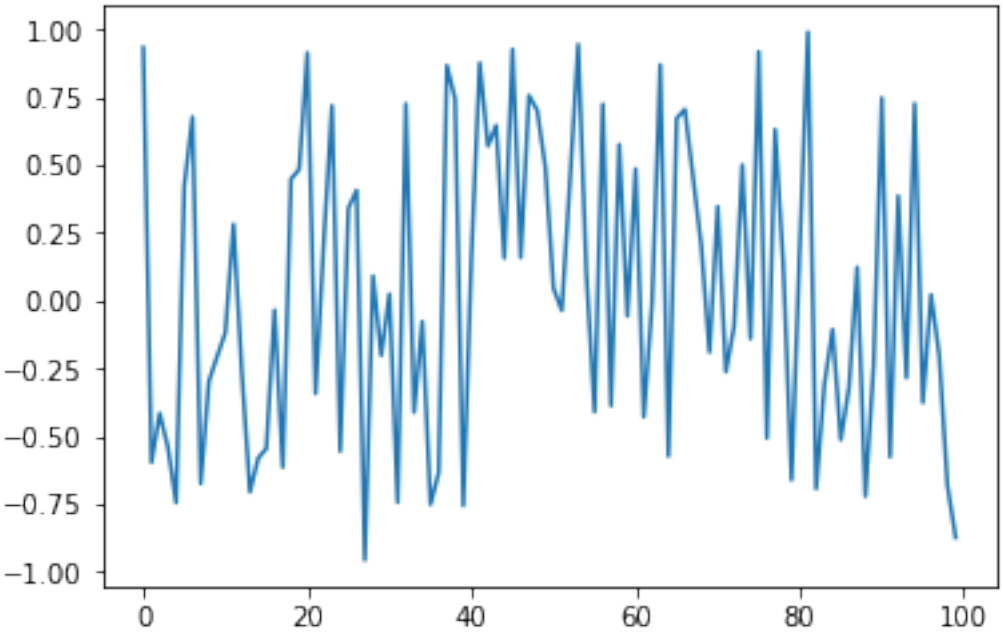
In [101]:
```
def wiener_process(period =100,n=1000,initial=0):
    t=period
    wp=[0]*t*n
    wp[0]=initial

    for i in range(1,t*n):
        wp[i]=wp[i-1]+random.gauss(0,1)/np.sqrt(n)
    return np.array(wp),t,n
```

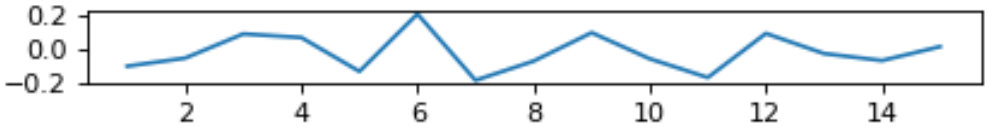
In [104]:
```
plot(random_walk())
acf_curve(random_walk())
```



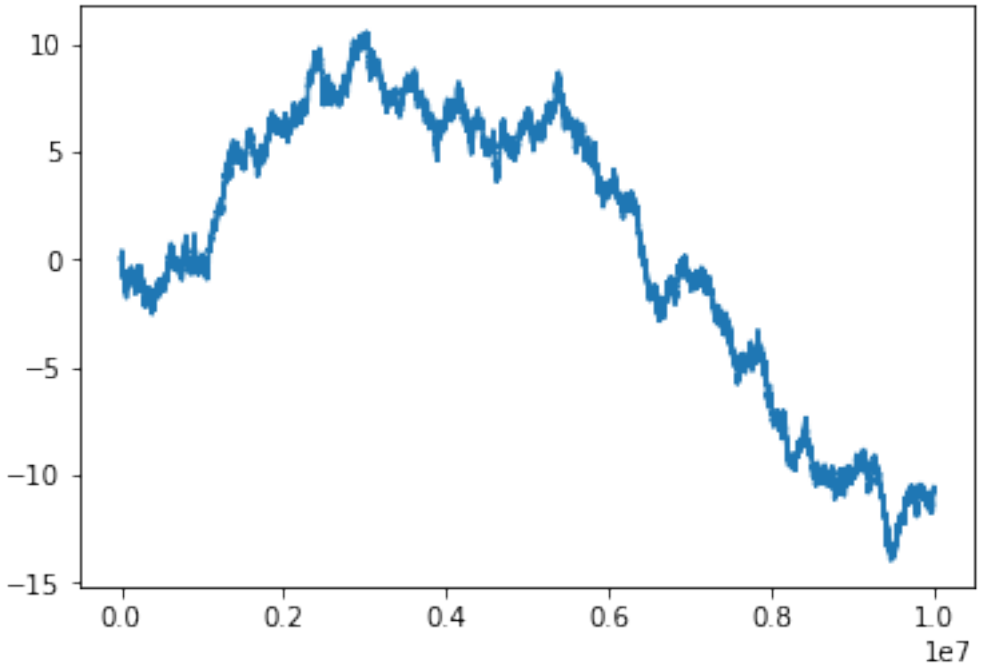


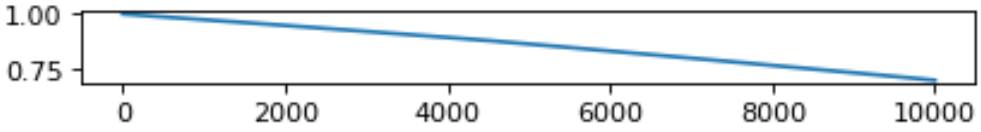
In [103]:
```
plot(white_noise())
acf_curve(white_noise())
```





In [102]:
```
wiener_process,t,n=wiener_process()
plot(wiener_process,t*n,step=1/t)
acf_curve(wiener_process,lag=10000)
```


10
5
0
−5
−10
−15
0.0
0.2
0.4
0.6
0.8
1.0
1e7


1.00
0.75
0
2000
4000
6000
8000
10000