In [358]:

```
import os
import pandas as pd
import numpy as np
import random

os.chdir('/Users/pbmrt/Desktop')
```

In [359]:

```
file = '/Users/pbmrt/Desktop/apple.2011.csv'
```

In [360]:

```
data = pd.read_csv(file,skiprows = [1])
```

In [361]:

```
data.columns = ['Date', 'Last', 'Chg']
```

In [362]:

```
data
```

Out[362]:

	Date	Last	Chg
0	1/4/2011	331.290	0.005219
1	1/5/2011	334.000	0.008180
2	1/6/2011	333.730	-0.000808
3	1/7/2011	336.120	0.007161
4	1/10/2011	342.455	0.018847
5	1/11/2011	341.640	-0.002380
6	1/12/2011	344.420	0.008137
7	1/13/2011	345.680	0.003658
8	1/14/2011	348.480	0.008100
9	1/18/2011	340.650	-0.022469
10	1/19/2011	338.840	-0.005313
11	1/20/2011	332.680	-0.018180
12	1/21/2011	326.720	-0.017915
13	1/24/2011	337.450	0.032842

14	1/25/2011	341.400	0.011705
15	1/26/2011	343.850	0.007176
16	1/27/2011	343.210	-0.001861
17	1/28/2011	336.100	-0.020716
18	1/31/2011	339.320	0.009580
19	2/1/2011	345.030	0.016828
20	2/2/2011	344.320	-0.002058
21	2/3/2011	343.440	-0.002556
22	2/4/2011	346.500	0.008910
23	2/7/2011	351.880	0.015527
24	2/8/2011	355.200	0.009435
25	2/9/2011	358.160	0.008333
26	2/10/2011	354.540	-0.010107
27	2/11/2011	356.850	0.006515
28	2/14/2011	359.180	0.006529
29	2/15/2011	359.900	0.002005
30	2/16/2011	363.130	0.008975
31	2/17/2011	358.300	-0.013301
32	2/18/2011	350.560	-0.021602
33	2/22/2011	338.610	-0.034088

In [363]:

```
mu = np.mean(data['Chg'])
```

In [364]:

```
sigma = np.std(data['Chg'])
```

In [365]:

```
px = np.array(data['Last'])
```

In [366]:

```
px
```

Out[366]:

```
array([ 331.29 ,  334.   ,  333.73 ,  336.12 ,  342.455,  341.64 ,
        344.42 ,  345.68 ,  348.48 ,  340.65 ,  338.84 ,  332.68 ,
        326.72 ,  337.45 ,  341.4  ,  343.85 ,  343.21 ,  336.1  ,
        339.32 ,  345.03 ,  344.32 ,  343.44 ,  346.5  ,  351.88 ,
        355.2  ,  358.16 ,  354.54 ,  356.85 ,  359.18 ,  359.9  ,
        363.13 ,  358.3  ,  350.56 ,  338.61 ,  342.62 ,  342.88 ,
        348.16 ,  353.21 ,  349.31 ,  352.12 ,  359.56 ,  360.   ,
        355.36 ,  355.76 ,  352.47 ,  346.67 ,  351.99 ,  353.56 ,
        345.43 ,  330.01 ,  334.64 ,  330.67 ,  339.3  ,  341.2  ,
        339.19 ,  344.97 ,  351.54 ,  350.44 ,  350.96 ,  348.63 ,
        348.507,  344.56 ,  341.19 ,  338.89 ,  338.04 ,  338.08 ,
        335.06 ,  330.8  ,  332.4  ,  336.13 ,  332.42 ,  327.46 ,
        331.85 ,  337.86 ,  342.41 ,  350.696,  353.009,  350.42 ,
        350.15 ,  346.75 ,  350.13 ,  346.28 ,  348.2  ,  349.57 ,
        346.75 ,  346.66 ,  347.6  ,  349.45 ,  347.23 ,  346.57 ,
        340.5  ,  333.3  ,  336.14 ,  339.87 ,  340.53 ,  335.22 ,
        334.4  ,  332.19 ,  336.78 ,  335.   ,  337.41 ,  347.83 ,
        345.51 ,  346.102,  343.44 ,  338.04 ,  332.04 ,  332.24 ,
        331.49 ,  325.9  ,  326.6  ,  332.44 ,  326.75 ,  325.16 ,
        320.26 ,  315.32 ,  325.3  ,  322.61 ,  331.23 ,  326.35 ,
        332.04 ,  335.26 ,  334.04 ,  335.67 ,  343.26 ,  349.43 ,
        351.76 ,  357.2  ,  359.71 ,  354.   ,  353.75 ,  358.02 ,
        357.77 ,  364.92 ,  373.8  ,  376.85 ,  386.9  ,  387.29 ,
        393.3  ,  398.5  ,  403.41 ,  392.59 ,  391.82 ,  390.48 ,
        396.75 ,  388.91 ,  392.57 ,  377.37 ,  373.62 ,  353.21 ,
        374.01 ,  363.69 ,  373.7  ,  376.99 ,  383.41 ,  380.48 ,
        380.44 ,  366.05 ,  356.03 ,  356.44 ,  373.6  ,  376.18 ,
        373.72 ,  383.58 ,  389.97 ,  389.99 ,  384.83 ,  381.03 ,
        374.05 ,  379.74 ,  383.93 ,  384.14 ,  377.48 ,  379.94 ,
        384.62 ,  389.3  ,  392.96 ,  400.5  ,  411.63 ,  413.45 ,
        412.14 ,  401.82 ,  404.3  ,  403.17 ,  399.26 ,  397.01 ,
        390.57 ,  381.32 ,  374.6  ,  372.5  ,  378.25 ,  377.37 ,
        369.8  ,  388.81 ,  400.29 ,  402.19 ,  408.43 ,  422.   ,
        419.99 ,  422.24 ,  398.62 ,  395.31 ,  392.87 ,  405.77 ,
        397.77 ,  400.6  ,  404.69 ,  404.95 ,  404.78 ,  396.51 ,
        397.41 ,  403.07 ,  400.24 ,  399.73 ,  406.23 ,  395.28 ,
        385.22 ,  384.62 ,  379.26 ,  388.83 ,  384.77 ,  377.41 ,
        374.94 ,  369.01 ,  376.51 ,  366.99 ,  363.57 ,  376.12 ,
        373.2  ,  382.2  ,  387.93 ,  389.7  ,  393.01 ,  390.95 ,
        389.09 ,  390.66 ,  393.62 ,  391.84 ,  388.81 ,  380.19 ,
        378.94 ,  381.02 ,  382.21 ,  395.95 ,  396.445,  398.55 ,
        403.33 ,  406.53 ,  402.64 ,  405.12 ,  405.   ])
```

In [367]:

```
rev_px = np.array(px[::-1])
```

In [368]:

```
rev_px
```

Out[368]:

```
array([ 405.   ,  405.12 ,  402.64 ,  406.53 ,  403.33 ,  398.55 ,
        396.445,  395.95 ,  382.21 ,  381.02 ,  378.94 ,  380.19 ,
        388.81 ,  391.84 ,  393.62 ,  390.66 ,  389.09 ,  390.95 ,
        393.01 ,  389.7  ,  387.93 ,  382.2  ,  373.2  ,  376.12 ,
        363.57 ,  366.99 ,  376.51 ,  369.01 ,  374.94 ,  377.41 ,
        384.77 ,  388.83 ,  379.26 ,  384.62 ,  385.22 ,  395.28 ,
        406.23 ,  399.73 ,  400.24 ,  403.07 ,  397.41 ,  396.51 ,
        404.78 ,  404.95 ,  404.69 ,  400.6  ,  397.77 ,  405.77 ,
        392.87 ,  395.31 ,  398.62 ,  422.24 ,  419.99 ,  422.   ,
        408.43 ,  402.19 ,  400.29 ,  388.81 ,  369.8  ,  377.37 ,
        378.25 ,  372.5  ,  374.6  ,  381.32 ,  390.57 ,  397.01 ,
        399.26 ,  403.17 ,  404.3  ,  401.82 ,  412.14 ,  413.45 ,
        411.63 ,  400.5  ,  392.96 ,  389.3  ,  384.62 ,  379.94 ,
        377.48 ,  384.14 ,  383.93 ,  379.74 ,  374.05 ,  381.03 ,
        384.83 ,  389.99 ,  389.97 ,  383.58 ,  373.72 ,  376.18 ,
        373.6  ,  356.44 ,  356.03 ,  366.05 ,  380.44 ,  380.48 ,
        383.41 ,  376.99 ,  373.7  ,  363.69 ,  374.01 ,  353.21 ,
        373.62 ,  377.37 ,  392.57 ,  388.91 ,  396.75 ,  390.48 ,
        391.82 ,  392.59 ,  403.41 ,  398.5  ,  393.3  ,  387.29 ,
        386.9  ,  376.85 ,  373.8  ,  364.92 ,  357.77 ,  358.02 ,
        353.75 ,  354.   ,  359.71 ,  357.2  ,  351.76 ,  349.43 ,
        343.26 ,  335.67 ,  334.04 ,  335.26 ,  332.04 ,  326.35 ,
        331.23 ,  322.61 ,  325.3  ,  315.32 ,  320.26 ,  325.16 ,
        326.75 ,  332.44 ,  326.6  ,  325.9  ,  331.49 ,  332.24 ,
        332.04 ,  338.04 ,  343.44 ,  346.102,  345.51 ,  347.83 ,
        337.41 ,  335.   ,  336.78 ,  332.19 ,  334.4  ,  335.22 ,
        340.53 ,  339.87 ,  336.14 ,  333.3  ,  340.5  ,  346.57 ,
        347.23 ,  349.45 ,  347.6  ,  346.66 ,  346.75 ,  349.57 ,
        348.2  ,  346.28 ,  350.13 ,  346.75 ,  350.15 ,  350.42 ,
        353.009,  350.696,  342.41 ,  337.86 ,  331.85 ,  327.46 ,
        332.42 ,  336.13 ,  332.4  ,  330.8  ,  335.06 ,  338.08 ,
        338.04 ,  338.89 ,  341.19 ,  344.56 ,  348.507,  348.63 ,
        350.96 ,  350.44 ,  351.54 ,  344.97 ,  339.19 ,  341.2  ,
        339.3  ,  330.67 ,  334.64 ,  330.01 ,  345.43 ,  353.56 ,
        351.99 ,  346.67 ,  352.47 ,  355.76 ,  355.36 ,  360.   ,
        359.56 ,  352.12 ,  349.31 ,  353.21 ,  348.16 ,  342.88 ,
        342.62 ,  338.61 ,  350.56 ,  358.3  ,  363.13 ,  359.9  ,
        359.18 ,  356.85 ,  354.54 ,  358.16 ,  355.2  ,  351.88 ,
        346.5  ,  343.44 ,  344.32 ,  345.03 ,  339.32 ,  336.1  ,
        343.21 ,  343.85 ,  341.4  ,  337.45 ,  326.72 ,  332.68 ,
        338.84 ,  340.65 ,  348.48 ,  345.68 ,  344.42 ,  341.64 ,
        342.455,  336.12 ,  333.73 ,  334.   ,  331.29 ])
```

In [369]:

```
sample = []
days = 20
for i in range(days):
    sample.append(rev_px[i])
```

In [370]:

```
print(sample)
```

```
[405.0, 405.12, 402.63999999999999, 406.52999999999997, 403.32999999999998, 398.5
5000000000001, 396.44499999999999, 395.94999999999999, 382.20999999999998, 381.01
999999999998, 378.94, 380.19, 388.81, 391.83999999999997, 393.62, 390.66000000000
003, 389.08999999999997, 390.94999999999999, 393.00999999999999, 389.699999999999
99]
```

In [371]:

```
sample_asc = sample[::-1]
```

In [372]:

```
sample_asc
```

Out[372]:

```
[389.69999999999999,
 393.00999999999999,
 390.94999999999999,
 389.08999999999997,
 390.66000000000003,
 393.62,
 391.83999999999997,
 388.81,
 380.19,
 378.94,
 381.01999999999998,
 382.20999999999998,
 395.94999999999999,
 396.44499999999999,
 398.55000000000001,
 403.32999999999998,
 406.52999999999997,
 402.63999999999999,
 405.12,
 405.0]
```

In [373]:

```
N = 10000
lastpxs = []
for i in range(N):
    newpx = []
    for i in range(len(sample_asc)):
        rand = random.gauss(mu,sigma)
        newpx.append((sample_asc[i]*rand)+sample_asc[i])
        if i==19:
            lastpxs.append((sample_asc[i]*rand)+sample_asc[i])
```

In [374]:

```
print newpx
```

```
[390.70228352126492, 395.23500169933448, 372.54905373472849, 388.57039696521815,
387.91416202287502, 399.03803749603952, 385.39791770685338, 399.2179937129182, 38
6.76276525852052, 391.76024024151201, 380.71383495952875, 379.1118689726465, 399.
74064182033493, 398.87433062212307, 406.03890036537859, 389.3595886240866, 408.05
125032348559, 414.49807823820441, 409.96948128576918, 407.8926186084185]
```

In [375]:

```
print np.max(newpx)
print np.min(newpx)
```

```
414.498078238
372.549053735
```

In [376]:

```
print np.mean(newpx)
```

```
394.569922309
```

In [377]:

```
len(lastpxs)
```

Out[377]:

```
10000
```

In [377]:

In [377]:

```

```

In [378]:

```
lastpxs_sort = np.sort(lastpxs)
```

In [379]:

```
lastpxs_sort
```

Out[379]:

```
array([ 378.60486339,  379.65422098,  380.48996743, ...,  429.96093433,
        430.96828341,  432.08252886])
```

In [380]:

```
top_percentile = np.percentile(lastpxs_sort,1)
```

In [381]:

```
top_percentile
```

Out[381]:

```
389.7132200720672
```

In [382]:

```
var = top_percentile
print "The Value at Risk for AAPL Stock with a 99% Confidence Interval is $",var,
"which means that prices should fall below $",var,"over a 20 day holding period l
ess than or equal to 1% of the time for the random sampling of 20 day close price
s which had a mean of $",np.mean(lastpxs_sort),"a max of $",np.max(lastpxs_sort),
"and a min of $",np.min(sample_asc),"."
```

```
The Value at Risk for AAPL Stock with a 99% Confidence Interval is $ 389.71322007
2 which means that prices should fall below $ 389.713220072 over a 20 day holding
 period less than or equal to 1% of the time for the random sampling of 20 day cl
ose prices which had a mean of $ 405.295960184 a max of $ 432.082528862 and a min
 of $ 378.94 .
```

In [382]:

```

```

In [382]:

```

```

In [382]: