Andy AbuMoussa

Final Project

I will model the stock market... Consider the system with one consumer and one stock.

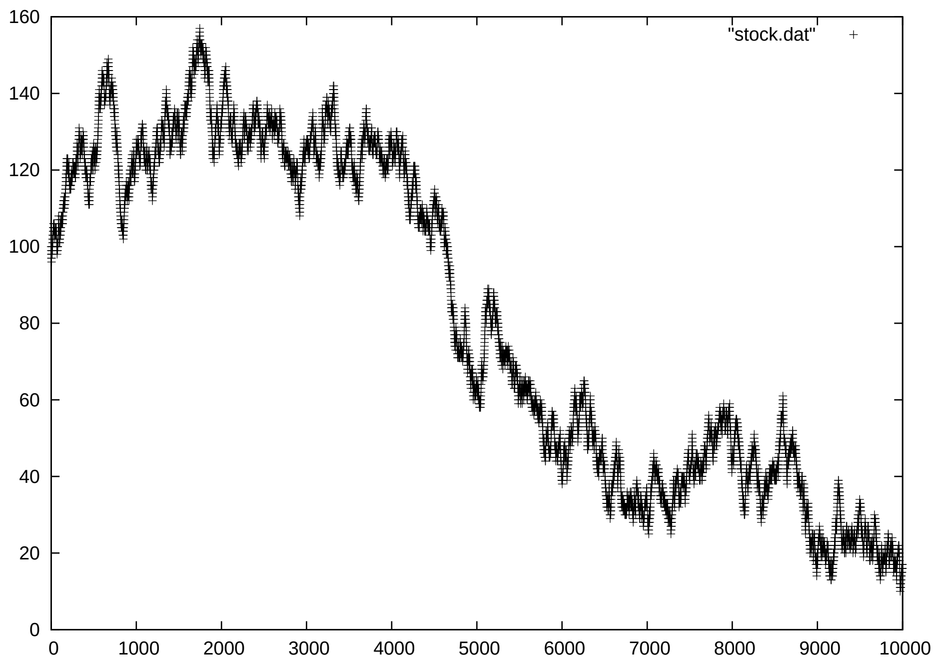
Construction of the stock: At time T_0, the stock starts at some initial price, denote it P_0 and has two outcomes at the end of one cycle. At the end of one cycle, the stock can either rise by one unit, or fall by one unit, let us say with probability of $1/2$. We will assume that the stock displays no activity during the cycle.

Completely Random Stock...

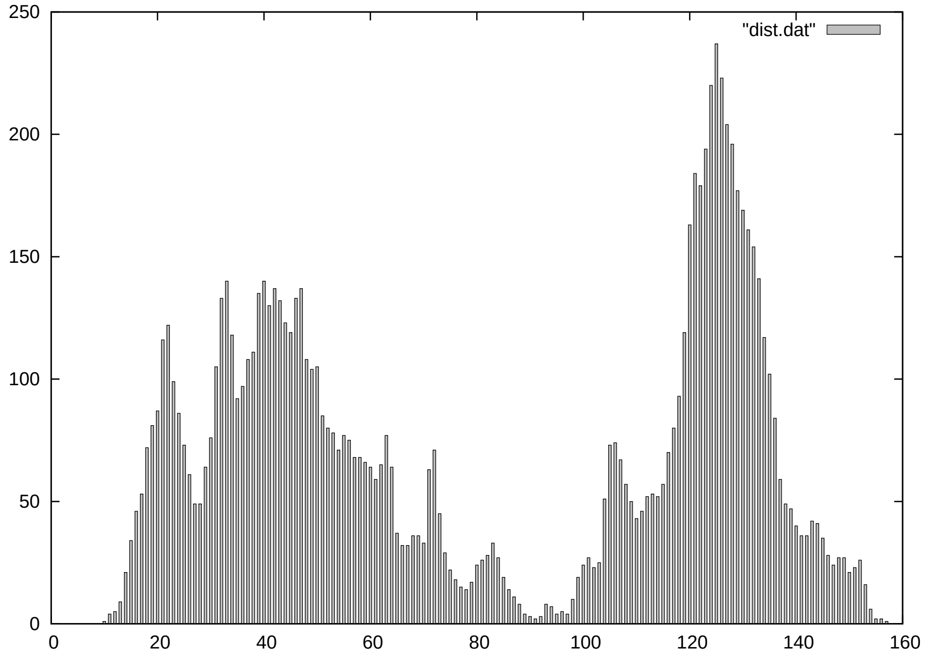
```
python] from random import random
python] from math import *
python] distribution=[0]
python] def stockPrice(p):
           if p==0:
               return 0
           if ceil(random()*1000000.)%2==0.:
               return p+1
           return p-1
python] def Distrib(p):
             if(len(distribution)>p): distribution[p]+=1
             else:
               for i in range(p-len(distribution)):
                  distribution.append(0)
               distribution.append(1)
python] def write2(file,distribution):
              for j in range(len(distribution)):
                 if distribution[j]!=0:
                   file.write(str(j)+' '+str(distribution[j])+'\n')
python] def write3(file,x,distribution):
              for j in range(len(distribution)):
                if distribution[j]!=0:
                   file.write(str(x)+' '+str(j+1)+' '+str(distribution[j])+'\n')
```

```
python] p=100.
python] stock=open('stock.dat','w')
python] for i in range(10000):
           stock.write(str(i)+' '+str(p)+'\n')
           Distrib(int(p))
           p=stockPrice(p)

python] stock.close();
python] dist=open('dist.dat','w')
python] write2(dist,distribution)

    None

python] dist.close();
python] multiD=open('multiD.dat','w')
python] for i in range(1000):
             p=100
             distribution=[0]
             for k in range(1000):
                Distrib(int(p))
                p=stockPrice(p)
             write3(multiD,i,distribution)
python] multiD.close();
python]
```

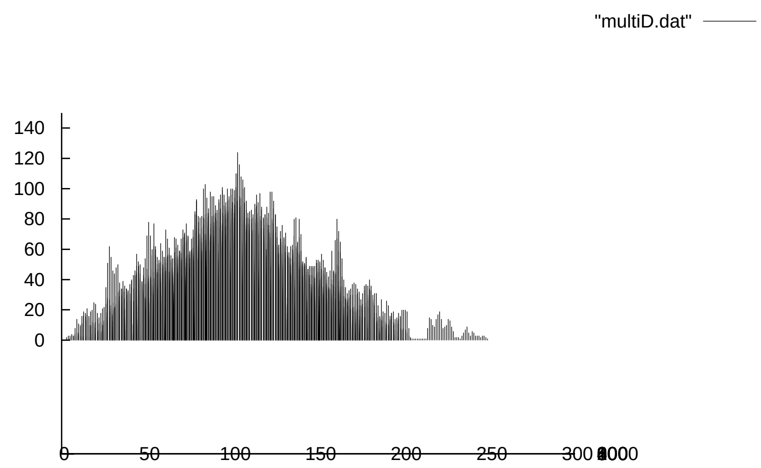
```
GNUplot] plot "stock.dat"
```



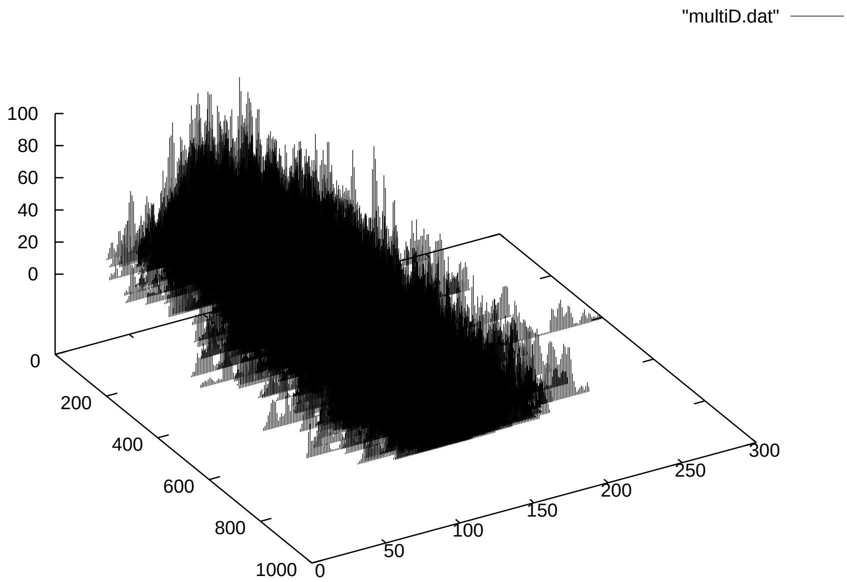
```
GNUplot] set style fill solid 0.25 border
         set boxwidth .5
         plot "dist.dat" with boxes
```




```
GNUplot] set view 90,90,1; set xr[0:1000];set yr [0:300];set zr[0:150];splot
         "multiD.dat" with boxes
```



```
GNUplot] set view 45,60,1; set xr[0:1000];set yr[0:300];set zr[0:100];splot
         "multiD.dat" with boxes
```




```
GNUplot]
```

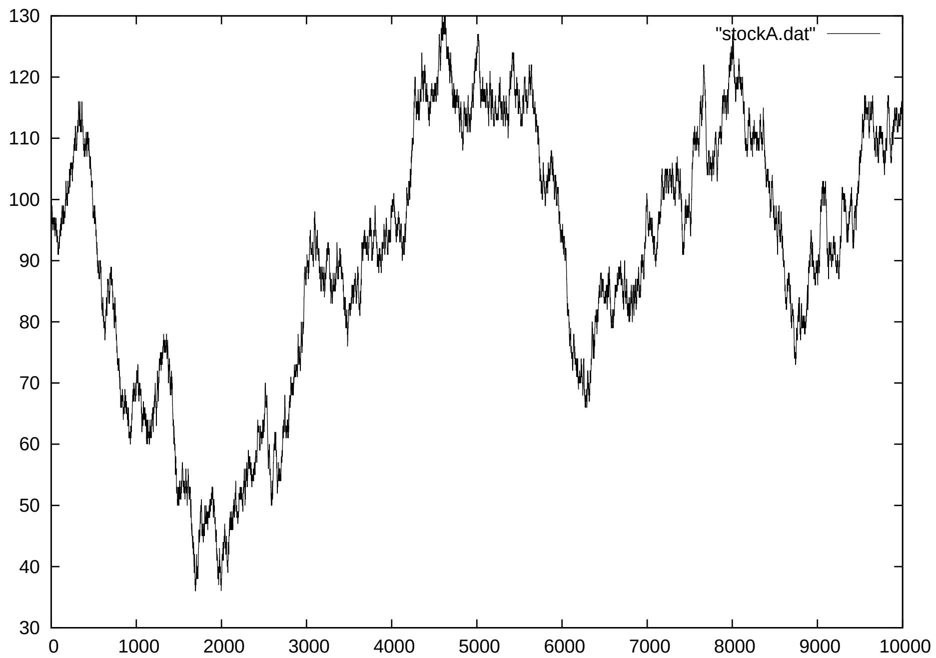
Construction of the stock: At time T_0, the stock starts at some initial price, denote it P_0 and has two outcomes at the end of one cycle. At the end of one cycle, the stock can either rise by one unit, or fall by one unit, let us say with probability of $1/2$. We will assume that the stock displays no activity during the cycle.

Completely Random Stock...

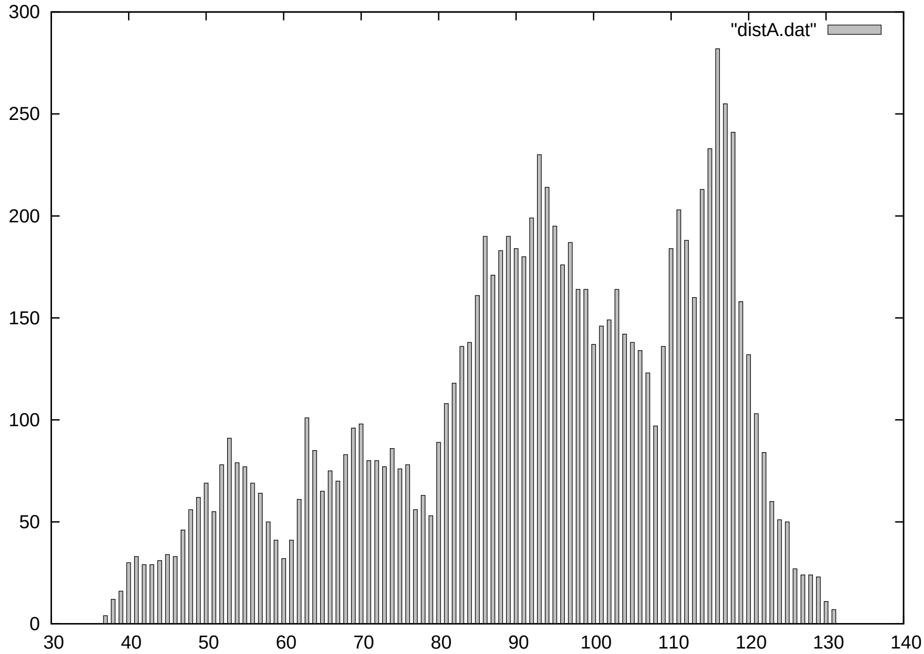
```
python] def stockPriceA(a):
           if a<=0:
               return 0
           x=ceil(random()*10000)%3.
           if x==1.:return a+1
           if x==2.:return a-1
           return a
```

```
python] p=100
python] distribution=[0]
python] stockA=open('stockA.dat','w')
python] for i in range(10000):
           stockA.write(str(i)+' '+str(p)+'\n')
           Distrib(p)
           p=stockPriceA(p)
python] stockA.close();
python] distA=open('distA.dat','w')
python] write2(distA,distribution)

    None

python] distA.close();
python] multiDA=open('multiDA.dat','w')
python] for i in range(1000):
            p=100
            distribution=[0]
            for k in range(1000):
               Distrib(p)
               p=stockPriceA(p)
            write3(multiDA,i,distribution)
python] multiDA.close();
python]
```

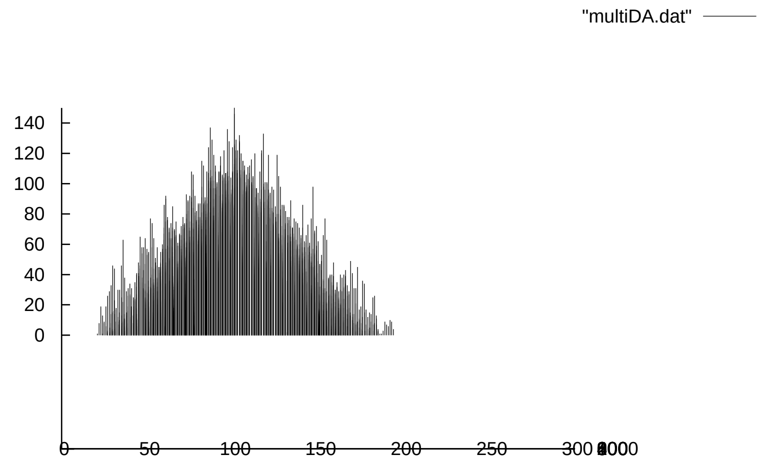
```
GNUplot] plot "stockA.dat" with lines
```



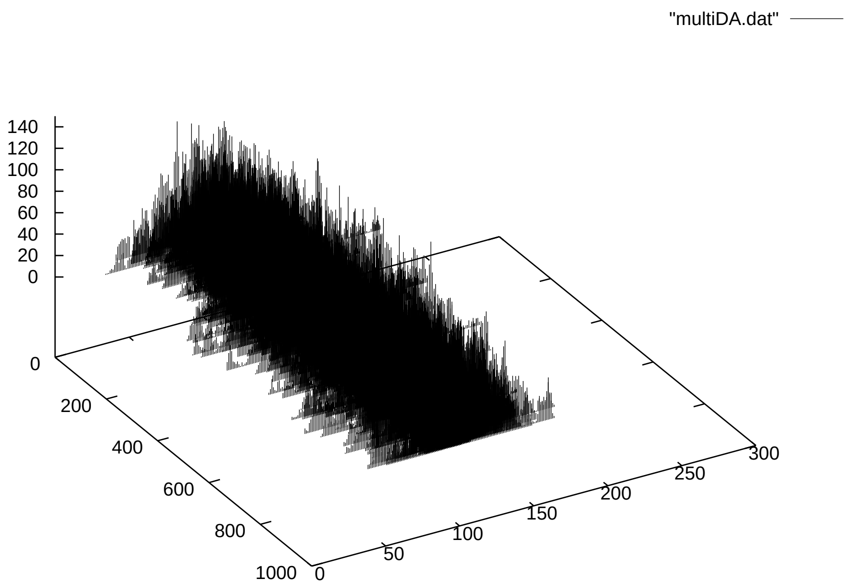
```
GNUplot] set style fill solid 0.25 border
         set boxwidth .5
         plot "distA.dat" with boxes
```




```
GNUplot] set view 90,90,1; set xr[0:1000];set yr [0:300];set zr[0:150];splot
         "multiDA.dat" with boxes
```




```
GNUplot] set view 45,60,1; set xr[0:1000];set yr[0:300];set zr[0:150];splot
         "multiDA.dat" with boxes
```




```
GNUplot]
```

Continuing the Construction of the Stock: Now, let us say that the stock has some artificial valuation, determined by the media, stock analysts, etc. Let's assume that the overall valuation of the stock is an average of all the valuations, and that it happens to be a constant function. We will call the valuation $P*$. If the price is within ε of $P*$, then the stock will only go up or down a fraction of the daily variaton during that cycle (we will say 1/2). I will simplify and say that $\varepsilon = 5$ for this next cycle.

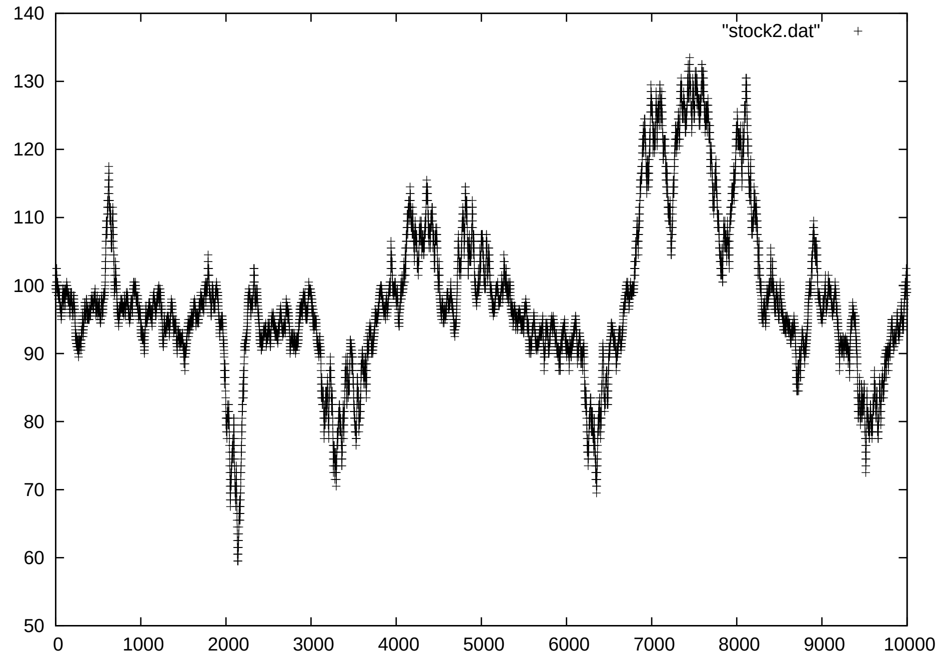
Random Stock with Constant Valuation...

```
python] from random import random
python] from math import *
python] def stockPrice2(p,ep,pstar,dep):
           if p<=0:
               return 0
           if ceil(random()*1000000.)%2==0.:
             if(abs(p-pstar)<=ep):
               return p+1./dep
             return p+1.
           if(abs(p-pstar)<=ep):
             return p-1./dep
           return p-1.
python] p=100.
python] ep=5.
python] dep=2.
python] pstar=95.
python] stock2=open('stock2.dat','w')
python] for i in range(10000):
           stock2.write(str(i)+' '+str(p)+'\n')
           p=stockPrice2(p,ep,pstar,dep)
python] stock2.close();
```

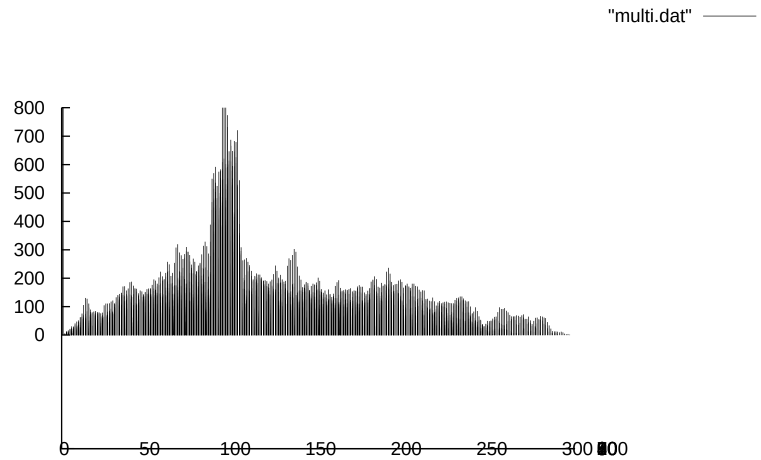
```
python] ## set up a multivariable function
python] multi=open('multi.dat','w')
python] for i in range(100):
          eps=i/10.
          dep=2.
          pstar=95.
          p=100.
          distribution=[0]
          for j in range(10000):
              Distrib(int(ceil(p)))
              p=stockPrice2(p,eps,pstar,dep)
          write3(multi,i,distribution)

python] multi.close();
python] multi2=open('multi2.dat','w')
python] for i in range(1000):
          eps=5
          pstar=95.
          p=100.
          dep=(i+1)/10.
          distribution=[0]
          for k in range(10000):
              Distrib(int(ceil(p)))
              p=stockPrice2(p,eps,pstar,dep)
          write3(multi2,i,distribution)
python] multi2.close();
python] def Distrib2(f,p):
             if(len(f)>p): f[p]+=1
             else:
               for i in range(p-len(f)):
                  f.append(0)
               f.append(1)
python] distance=open('distance.dat','w'); dp=[0];dm=[0];
python] for i in range(10000):
          eps=i/10.
          dep=2.
          pstar=95.
          p=100.
          for j in range(10000):
              p=stockPrice2(p,eps,pstar,dep)
          if(p-100>=0):Distrib2(dp,int(floor(p-100)))
          else:Distrib2(dm,int(ceil(abs(p-100))))
python] write2m(distance,dm)

    None

python] write2(distance,dp)

    None

python] distance.close();
python] dp

    [0]
```

```
python] def write2m(file,distribution):
              for j in range(len(distribution)):
                if distribution[len(distribution)-1-j]!=0:
                   file.write(str(-1*(len(distribution)-1-j))+'
        '+str(distribution[(len(distribution)-1-j)])+'\n')
python]
```

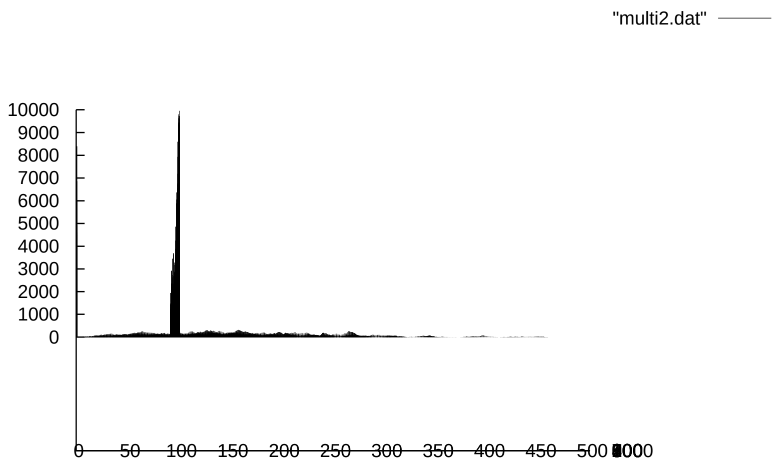
```
GNUplot] plot "stock2.dat"
```



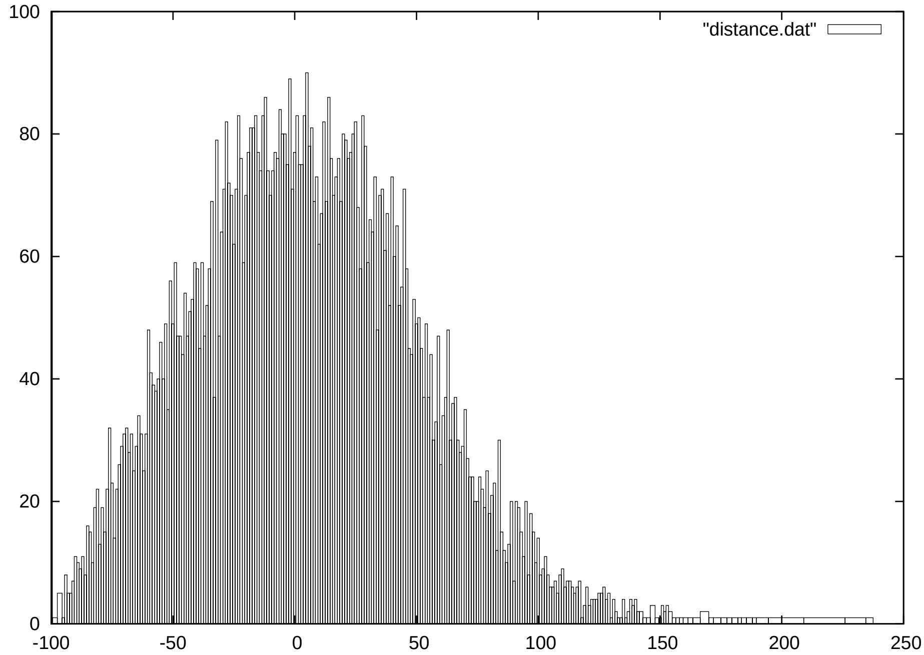

```
GNUplot] set view 90,90,1;set zr[0:800];splot "multi.dat" with boxes
```



```
GNUplot] set view 90,90,1;splot "multi2.dat" with boxes
```




```
GNUplot] set yr[0:100];plot "distance.dat" with boxes
```



GNUplot]

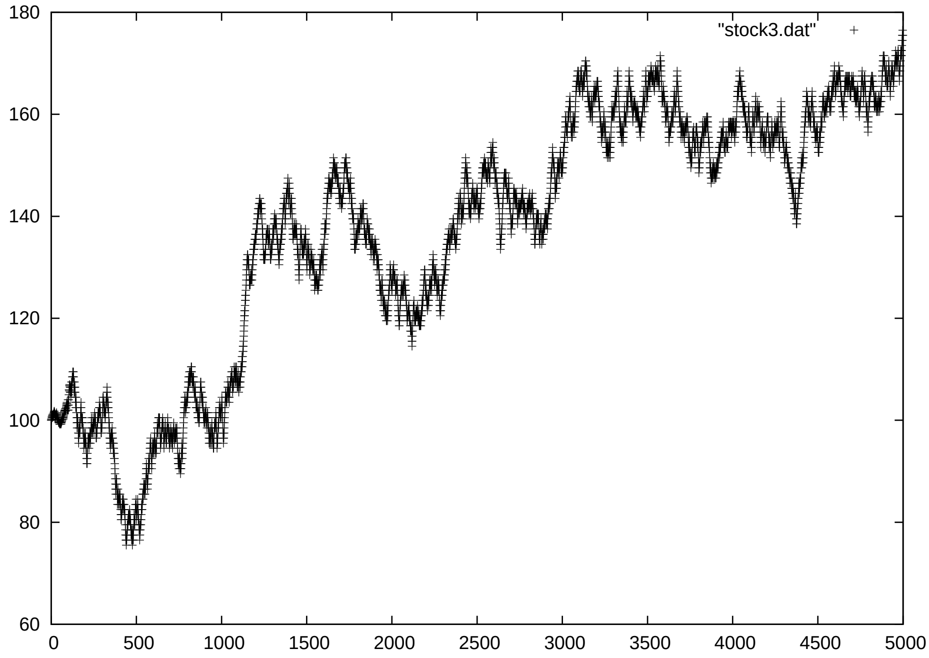
Continuing the construction of the Stock: Let us now consider the media, brokers, etc valuing the stock with some function, call it $f(t)$. This case will follow the same form as above, where some tolerance will be given to the valuation of the stock. Let us observe...

Random Stock with Time Variable Valuation...

```
python] from random import random
python] from math import *
python] def f(t): return .25*t+90
python] def ff(t): return .25
python] def g(t): return 5*cos(t)+95
python] def gg(t): return -5*sin(t)
python] def stockPrice3(p,ep,h,hh,t):
            if p<=0:
                return 0
            if ceil(random()*1000000.)%2==0.:
              if(abs(p-h(t))<=ep):
                return p+hh(t)
              return p+1.
            if(abs(p-f(t))<=ep):
              return p-hh(t)
            return p-1.
python] p=100.
python] p3=p
python] p4=p
python] stock3=open('stock3.dat','w')
python] stock4=open('stock4.dat','w')
python] for i in range(5000):
            ep=p3/10.
            stock3.write(str(i)+' '+str(p3)+'\n')
            p3=stockPrice3(p3,ep,f,ff,i)
            ep=p4/10.
            stock4.write(str(i)+' '+str(p4)+'\n')
            p4=stockPrice3(p4,ep,g,gg,i)
```

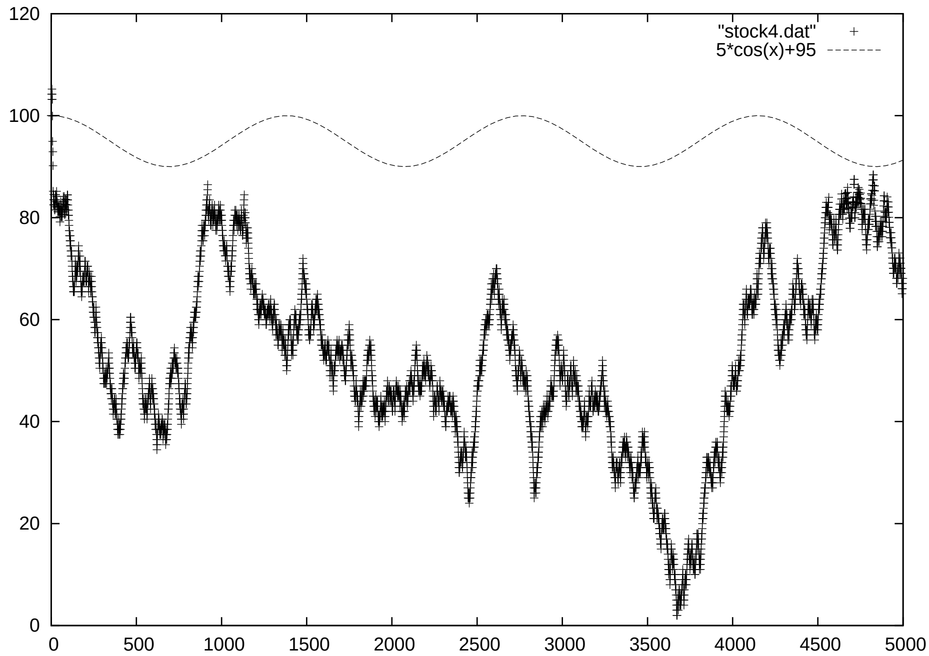
```
python] stock3.close();
python] stock4.close();
```

```
GNUplot] plot "stock3.dat" with lines  ##Linear Valuation
```



-ne

```
GNUplot] plot "stock4.dat", 5*cos(x)+95 ##Cosine Valuation
```




```
GNUplot]
```

Construction of the Stock: Consider now, a market with consumer driven psychological dependencies.

-Assumptions, if $P_{n+1} < P_n$ then $P_{n+2} < P_{n+1}$ with greater probability, vice versa

i.e. Consumers see trends in an otherwise random prosses.

Construction of this: Apply a linear fit to the last 50 data points.

If slope is within some tolerance, probability will remain random...

Else, slope of the line and tendencancy of the price affect consumer behavior.

Construction of the Consumer: We will assume that the consumer has capital at time T_0 equivalent to the value of price of the stock at that time. At the begining of each cycle, the consumer has the option of buying or selling the stock. We will say that the consumer wants to maximize profit given by $X = P(t) - C_0$ where $P(t)$ is the value of the stock at time t and C_0 is the capital the consumer has at time t_0.

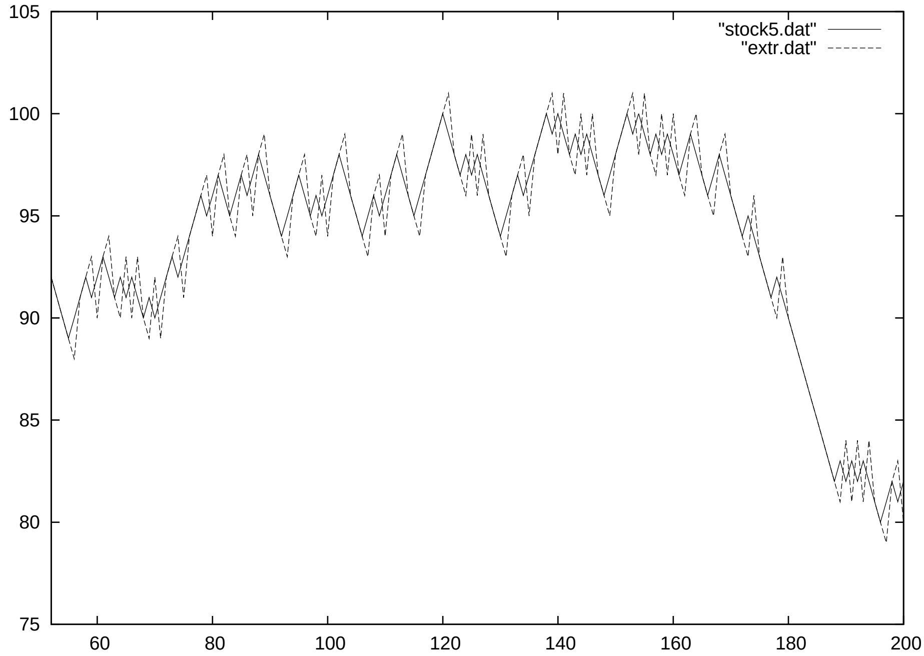
Part I - Construction of Consumer Psychology (2 step view)...

```
python] from random import random
python] from math import *
python] def extLag2(i):
           if i==0:
              return 2.*p10[9]-p10[8]
           if i==1:
              return 2.*p10[0]-p10[9]
           return 2.*p10[i-1]-p10[i-2]
python] def stockPrice4(p):
           if p<=0:
               return 0
           if ceil(random()*100.)%2==0.:
               return p+1.
           return p-1.
python] p=100.
python] pstar=105
python] p10=[0,0,0,0,0,0,0,0,0,0,0]
python] stock5=open('stock5.dat','w')
python] extr=open('extr.dat','w')
python] ep=1
python] for i in range(20000):
           p10[i%10]=ceil(floor(p*100))/100.
           stock5.write(str(i)+' '+str(p)+'\n')
           if(i>1):
              extr.write(str(i)+' '+str(extLag2((i)%10))+'\n')
           p=stockPrice4(p)

python] stock5.close();
python] extr.close();
```

python]

GNUplot]

```
i=50; set xr[2+i:150+i]; plot "stock5.dat" with lines, "extr.dat" with
lines
```



GNUplot]

Construction of a Stochastic Integral: Through my construction of the stock market, I found it difficult to apply any real theory to the behavior of this random market. The stock market being so complex and the subject of a great many number of dissertations, I decided to change the focus of my project. I thought that I had constructed a bunch of useless data, but what I had really generated was a mechanism by which I can propose a behavior for the "lazy consumer". Many argue that by arbitrage and other stradegies, the market removes any way for a consumer to earn easy money. I beg to differ.

Here's why:

Considering any stock and any time step of choice, there are really only three behaviors for a given stock. Let's quantify them by:

$$\begin{array}{ccl} & & P_1 = P_0 + \delta p \\ & \nearrow & \\ P_0 \longmapsto & \longrightarrow & P_1 = P_0 \\ & \searrow & \\ & & P_1 = P_0 - \delta p \end{array}$$

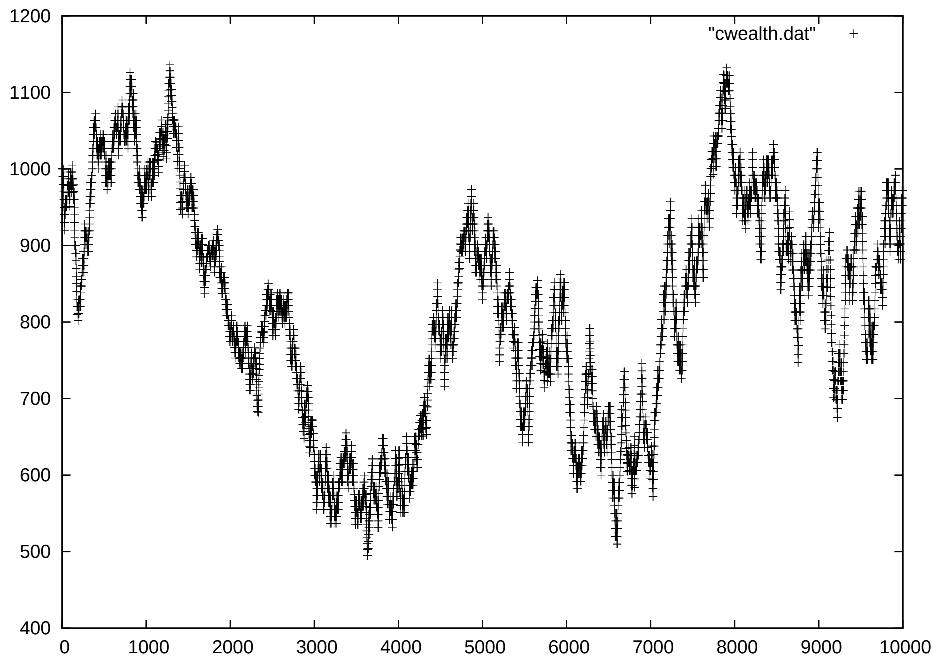
Now, let's say that the consumer has a simple trading stradegy... namely, if the stock goes down in price three times in a row, the consumer buys the stock, and if the stock goes up 4 times, he sells.

```
python] def market(w,com,u,d):
          wealth=w
          stocks=0
          up=0
          down=0
          p=100.
          for i in range(10000):
             cwealth.write(str(i)+' '+str(wealth+stocks*p)+'\n')
             x=stockPriceA(p)
             if(x>p):up+=1
             else: down+=1
             p=x
             if(up==u):
                wealth+=stocks*p-com
                stocks=0
                up=0
             if(down==d):
                buy=floor((wealth-com)/p)
                wealth-=buy*p+com
                stocks+=buy
                down=0
python] cwealth=open("cwealth.dat",'w')
python] market(1000,0,10,4)
```

```
    None
```

```
python] cwealth.close();
python]
```

```
GNUplot] plot "cwealth.dat"
```

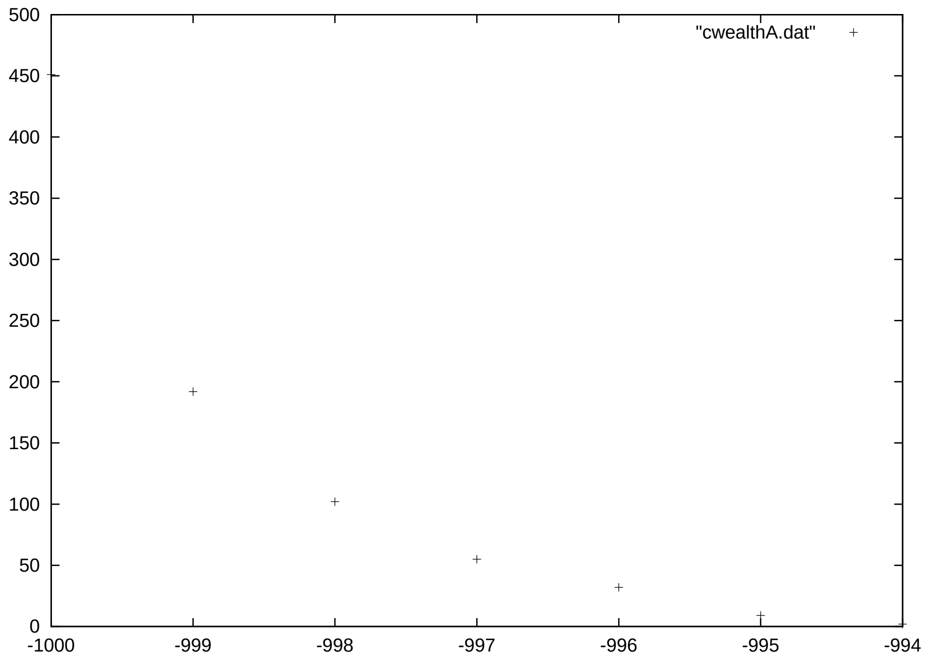


GNUplot]

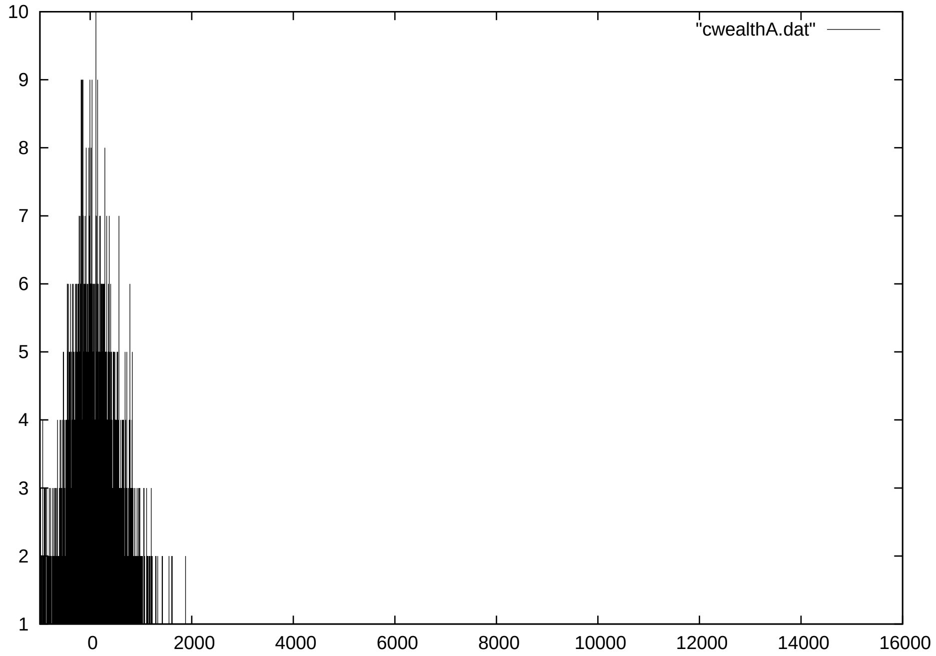
```
python] def marketDist(w,com,u,d):
          distribution=[0]
          total=0
          totaln=0
          zeroes=0
          dist=[0]
          distn=[0]
          for j in range(5000):
           wealth=w
           stocks=0
           up=0
           down=0
           p=100.
           for i in range(10000):
             x=stockPriceA(p)
             if(x>p):up+=1
             else: down+=1
             p=x
             if(up==u):
                wealth+=stocks*p-com
                stocks=0
                up=0
             if(down==d and p!=0):
                buy=floor((wealth-com)/p)
                wealth-=buy*p+com
                stocks+=buy
                down=0
           if(ceil(wealth+stocks*p)>w):
            total+=1
            add(dist,int(ceil(wealth+stocks*p)-w))
            continue
           if(ceil(wealth+stocks*p)<w):
            totaln+=1
            add(distn,int(w-ceil(wealth+stocks*p)))
            continue
           zeroes+=1
           add(dist,0)
          write32(cwealthA,distn)
          write33(cwealthA,dist)
          print "you lose money", totaln/50., "percent of the time"
          print "you earn money", total/50., "percent of the time"
```

```
python] def add(x,j):
             if(len(x)>j): x[j]+=1
             else:
               for i in range(j-len(x)):
                  x.append(0)
               x.append(1)
python] def write32(file,neg):
           for j in range(len(neg)):
                h=len(neg)-j-1
                if neg[h]!=0:
                   file.write(str(-h)+' '+str(neg[h])+'\n')
python] def write33(file,pos):
              for j in range(len(pos)):
                if pos[j]!=0:
                   file.write(str(j)+' '+str(pos[j])+'\n')
python] cwealthA=open("cwealthA.dat",'w')
python] marketDist(1000,0,4,10)

   you lose money 52.78 percent of the time
   you earn money 47.18 percent of the time
```

```
    None

python] cwealthA.close();
python]
python]

GNUplot] set xr[-1000:-994];plot "cwealthA.dat"
```




```
GNUplot] set xr[-990:16000];plot "cwealthA.dat" with impulses
```




```
GNUplot]
```

```
python] from math import *
python] p2=0
python] def logApp():
           p2=100.
           p1=100.
           p=100.
           cwealth=10,000
           for i in range(10000):
              p2=p1
              p1=p
              p=stockPriceA(p)
              x=log(p2)/log(p1)
              if (x<.85):return(p/p1)
python] logApp()
    None
python]
python]
```