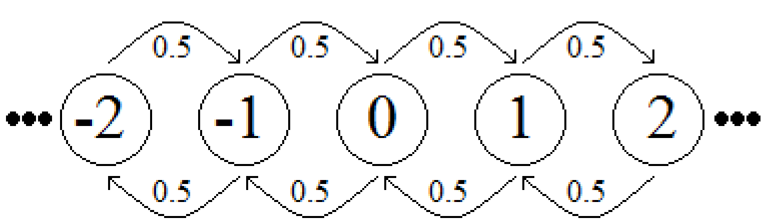
Random Walks in 1D and 2D

Tyler Paul

September 20, 2013

1 Introduction

A random walk is a type of stochastic process. A stochastic process is a set of random variables indexed by a set corresponding to time; that is $\{X_t : t \in T\}$ [1]. We will take T to be the set of nonnegative integers. Each random variable in the process can be thought of as possible states of a particle. Particularly, the random variable X_0 is the initial possible states of the particle and X_t is the possible states of the particle after t units of time. We will have the random variable take on integer values. The initial state in our examples can be thought of as a particle being fixed at the origin; that is $P(X_0 = 0) = 1$. Now as the defining feature of a random walk, when a unit of time passes, the particle has a 50% probability of moving to the right one unit and a 50% probability of moving to the left one unit. This random walk process is represented in the figure below. Hence $P(X_1 = 1) = 0.5$ and $P(X_1 = -1) = 0.5$. As for the second time step, since there is a .5 probability of the particle being at position 1 and there is a 0.25 probability of the particle at position 1 to move to position 2 and a 0.25 probability of moving from position 1 back to position 0. Similarly, since there is a .5 probability of the particle being at position -1 and there is a 0.25 probability of the particle at position -1 to move to position -2 and a 0.25 probability of moving from position 1 back to position 0. Hence overall: $P(X_2 = -2) = P(X_2 = 2) = 0.25$ and $P(X_2 = 0) = 0.5$. This process continues indefinitely.



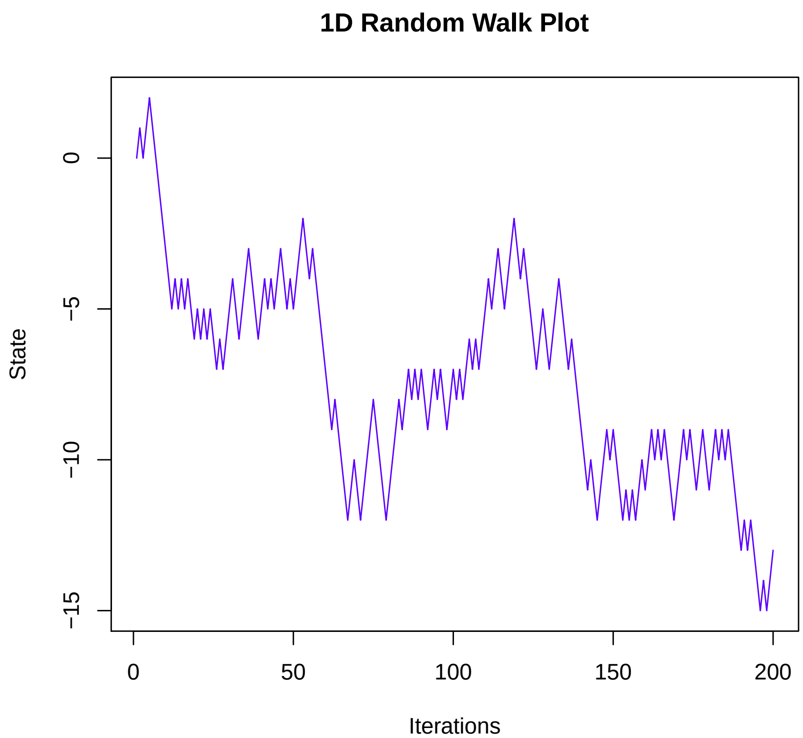
2 1D Random Walk

We now implement a sequence of R commands which generate a 1D random walk. To do so we use the *sample* command to choose the number 1 or 2 randomly (which corresponds to a 50/50 probability). Based on this value we then either increment or decrement the state of the particle by 1 (and using an initial state of 0). We then repeat this process the desired amount of iterations. In our example we use 200 iterations. Thus we use the code:

```
num=200 #max number of iterations
x=0 #vector to hold the states of the particle
state=0
```

```
x[1]=state
for(i in 2:num)
{
    s=sample(1:2,1)
    if(s==1){
        state=state+1
    }
    if(s==2){
        state=state-1
    }
    x[i]=state #save state
}
plot(1:200,x,type="l")
```

This code generates the random walk shown below.



3 PMF for a 1D Random Walk

One approach to determing the probability mass function for a particular time step (denoted p_t) of a 1D random walk is a combinatorial approach. Let $t \in T$ and suppose we want to find $p_t(x)$. For the particle to move from position 0 to position x in t time steps, the particle must move r units to the right and l units to the left where $r + l = t$ and $x = r - l$. Note that there are $\binom{r+l}{r}$ possible ways to choose a path which contains a total of $r + l$ steps and r steps to the right. Since $r + l = t$ then there are $\binom{t}{r}$ possible paths. Additionally, the probability that a single path using r steps to the right is chosen is $(0.5)^r(0.5)^l = (0.5)^r(0.5)^{t-r}$. Hence

$$p_t(x) = \binom{t}{r}(0.5)^r(0.5)^{t-r}$$

where $r = \frac{t+x}{2}$. Oberserve that the right hand side is the binomial distribution (in r, not in x) with $p = 0.5$.

However, note that $p_t(x)$ does not make sense for all x. For example, if $t = 3$ and $x = 2$, then $r = \frac{5}{2}$. Clearly, this is nonsense since our model only makes use of integer steps to the right. In fact, $p_3(2) = 0$. There are 8 possible paths using 3 time steps:

$$\begin{array}{ll}
0 \to 1 \to 0 \to 1 & 0 \to 1 \to 0 \to -1 \\
0 \to 1 \to 2 \to 3 & 0 \to 1 \to 2 \to 1 \\
0 \to -1 \to -2 \to -1 & 0 \to -1 \to -2 \to -3 \\
0 \to -1 \to 0 \to 1 & 0 \to -1 \to 0 \to -1
\end{array}$$

yet no path results in a state of 2. Generalizing this idea, $p_t(x) = 0$ when $r = \frac{t+x}{2} \notin \mathbb{Z}$.

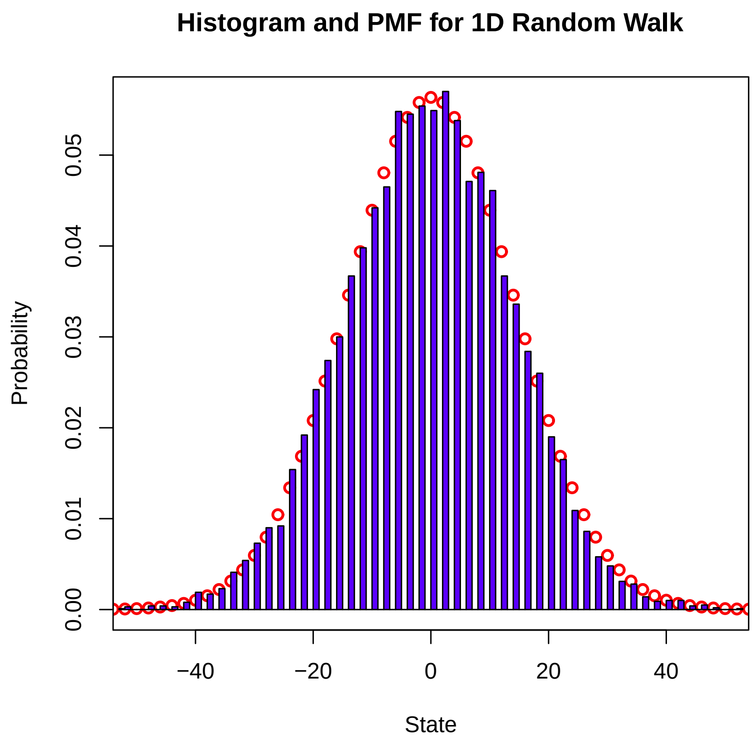
4 PMF Comparison using R

We can use R to run the code in section 1 for numerous iterations (we use 10,000 in our example) noting the position of the particle after 200 times steps (or for any t value one would like) for each iteration. We can then construct a histogram and superimpose the PMF p_{200} to see how well they match.

```
x=0
v=0
num=200
maxIterations=10000
for(j in 1:maxIterations)
{
    x=0
    state=0
    x[1]=state
    for(i in 2:num)
    {
        s=sample(1:2,1)
        if(s==1){
            state=state+1
        }
        if(s==2){
            state=state-1
        }
    }
    v[j]=state
}
x=seq(-num,num,by=2)
```

```
y=dbinom((x+num)/2, size=num, prob=.5)
plot(x,y,lwd=2,col="red",xlim=c(-50,50),xlab="State",ylab="Probability",
+ main="Histogram and PMF for 1D Random Walk")
hist(v,breaks=90,freq=F,col="blue",add=T)
```

This code yields the following plot. Note that there are spaces between the bars in the histogram. This is due to the fact that $p_{200}(x)$ is 0 when x is odd since this means that $\frac{200+x}{2} \notin \mathbb{Z}$.



5 R Computation of Statistical Information

We can use R to calculate quantities such as the five number summary and standard deviation corresponding to the final states of the particle. We can use the code below which calculates the statistical information regarding the final state of the particle after 200 steps (recall that the vector v from section 4 holds the data regarding the states).

```
summary(v)
sd(v)
```

The summary yields

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-51.0000  -9.0000  -1.0000  -0.0318   9.0000  49.0000
```

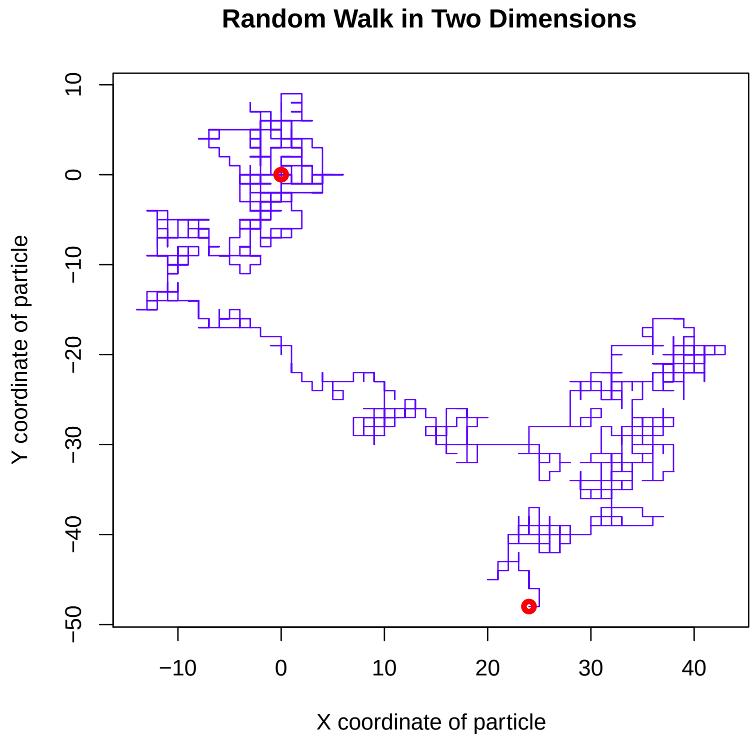
and the standard deviation is 14.07226. Note that we can rerun the code using a different number of time steps if desired.

6 2D Random Walk

We can extend the concept of random walks to higher dimensions. We can slightly adapt the code used in section 1 to generate a 2D random walk. In this case the state of a particle is specified by an element of $\mathbb{Z}^2$. We set the initial state of the particle to be at $(0,0)$. Additionally, we implement that there is a 0.25 probability each that the particle will move up, down, right, or left. To implement this idea and generate a plot of the position of the particle at each point in the walk we use the following code.

```
x=0
y=0
state1=0
state2=0
x[1]=state1
y[1]=state2
iterations=1500
for(i in 1:iterations)
{
    s=sample(1:4,1)
    if(s==1){
        state1=state1+1
        state2=state2
    }
    else if(s==2){
        state1=state1
        state2=state2+1
    }
    else if(s==3){
        state1=state1-1
        state2=state2
    }
    else{
        state1=state1
        state2=state2-1
    }
    x[i]=state1 #save state
    y[i]=state2
    i=i+1
}
plot(x,y,main="Random Walk in Two Dimensions",xlab="X coordinate of particle",
+ ylab="Y coordinate of particle",type="l",col="blue")
points(x[length(x)],y[length(y)],col="red",lwd=4)
points(0,0,col="red",lwd=4,add=T)
```

This code generates the plot below. The initial and final point of the random walk are specified by the red dots.



7 Conclusion

From the above information, we have written R code to generate random walks in both 1 and 2 dimensions. Additionally, we have found the probability mass function for the 1D case for any particular t value. It was also shown that the PMF for a particular t value has alternating locations of 0 probability. If t is odd, then $p_t(x) = 0$ whenever x is even. Similarily, if t is even, then $p_t(x) = 0$ whenever x is odd. Lastly, R can be used to calculate statistical quantities such as the five number summary and standard deviation of the final state of the particle after a given number of time steps.

References

[1] Logan, David. *Applied Mathematics Third Edition*. New Jersey: John Wiley & Sons Publication, 2006. Print.