# EAC 2

# Probabilidad y Estadística


| | |
|---|---|
| De Elias, Miguek | 50457 |
| Ordano, Esteban | 50753 |
| Crespo, Alvaro | 50758 |


26 de mayo de 2011

# Enunciado

En la caminata aleatoria o *random walk* (ó *camintata del borracho*) unidimensional (ó 1D) se considera el movimiento en una dimensión (a lo largo de una línea recta) de un punto que ocupa posiciones de abscisa entera. En cada instante (consideramos intervalos regulares) el punto se mueve a la derecha (incrementando su posición en una unidad) con probabilidad $p$ ó hacia la izquierda con probabilidad (1 - $p$), en este caso la coordenada de posición se reduce en unidad.
Si se asume que el movimiento es 2D entonces las coordenadas $x$ e $y$ se consideran enteras y en cada instante a partir de la posición con coordenadas ($x$,$y$) con distribución uniforme la próxima posición puede tomar uno de estos 4 valores: ($x+1$,$y$),($x-1$,$y$),($x$,$y+1$) ó ($x$,$y-1$). En la versión 3D las coordenadas $x$, $y$ y $z$ se consideran enteras y en cada instante a partir de la posición con coordenadas ($x$,$y$,$z$) y con distribución uniforme la próxima posición puede tomar uno de estos 6 valores: ($x+1$,$y$, $z$),($x-1$,$y$,$z$),($x$,$y+1$,$z$), ($x$,$y-1$,$z$), ($x$,$y$,$z+1$) ó ($x$,$y$,$z-1$) .
Suponga que $p = 0{,}5$ en el caso 1D y que la posición inicial en cualquiera de los casos es el origen de coordenadas.

a) Estime la probabilidad de la *vuelta a casa* en la caminata aleatoria 1D simétrica ($p = 0{,}5$) por medio de una simulación. Considere el transcurso del tiempo (*paso a paso*) hasta que la coordenada de posición vuelva a valer cero. Suponga como máximo un número grande de posibles transiciones para creer en el no retorno (por ejemplo 10000 pasos). Considere un número grande de pruebas ( también 10000 ó más). Con la información disponible estime la probabilidad pedida y también el número promedio de pasos hasta la vuelta. Haga un histograma para el número de pasos.
Tome alguna trayectoria (sucesión de posiciones) que haya resultado *larga* hasta que se produce el retorno (tal vez limitando el número máximo de pasos considerado) y represéntela con la posición y el tiempo como las coordenadas de un gráfico de dispersión.

b) Estime la probabilidad de la *vuelta a casa* en la caminatas aleatorias 2D y 3D por medio de una simulación. Considere el transcurso del tiempo (*paso a paso*) hasta que la coordenada de posición vuelva a ser el vecto nulo. Suponga como máximo un número grande de posibles transiciones para creer en el no retorno (por ejemplo 10000 pasos). Tome alguna trayectoria (sucesión de posiciones) que haya resultado *larga* y represéntela en 2D para la caminata 2D y en 3D para la otra.

Algunas referencias de recomendable consulta para poder entender algo de lo que vaya a observar en la simulación:

1. http://mathworld.wolfram.com/RandomWalk1-Dimensional.html
2. http://mathworld.wolfram.com/RandomWalk2-Dimensional.html
3. http://mathworld.wolfram.com/RandomWalk3-Dimensional.html
4. http://mathserver.sdu.edu.cn/mathency/math/p/p447.htm

# 1. Punto A

Utilizando el código del apéndice XXXXXXXXX (con dim = 1), para la caminata aleatoria 1D, y considerando un número de pasos igual a 10000 y de simulaciones igual a 10000,los resultados arrojados fueron:

Probabilidad de la vuelta a casa: 0.9914.
Promedio pasos hasta la vuelta: 75.55.

A continuación observamos el histograma para el número de pasos. En él se puede observar la cantidad de simulaciones que lograron volver exitosamente para cada grupo de número de pasos.

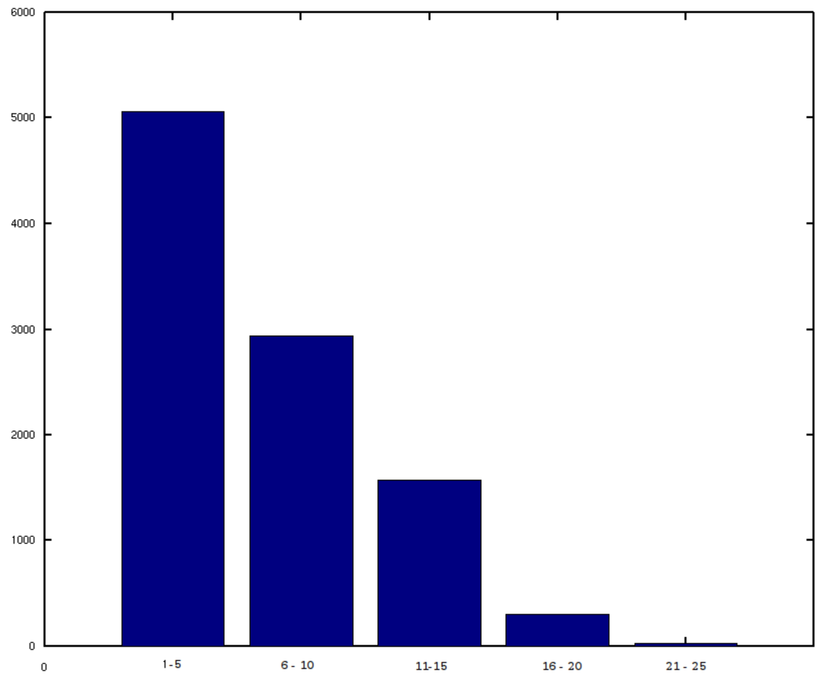


**Figura 1:** Histograma

En la siguiente figura, se puede ver la trayectoria de una de las caminatas aleatorias que logro exitosamente volver a casa.

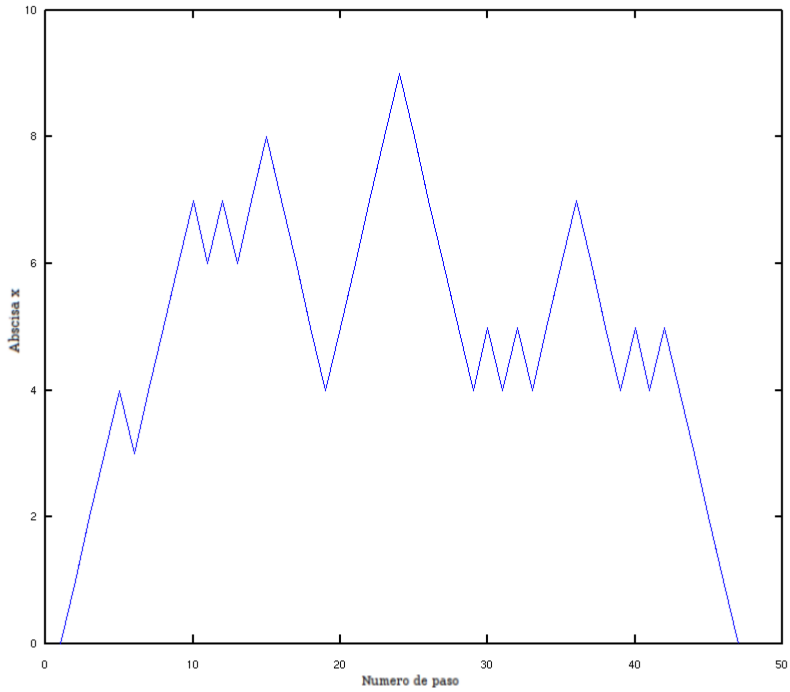


**Figura 2:** Trayectoria de una caminata aleatoria 1D

## 2. Punto B

### Caminata aleatoria 2D

Para la caminata aleatoria 2D, utilizando de nuevo el mismo código del apéndice XXXXX (esta vez con dim = 2), y considerando un número de pasos igual a 10000 y de simulaciones igual a 10000,los resultados arrojados fueron:

Probabilidad de la vuelta a casa: 0.7350.
Promedio de pasos hasta la vuelta: 366.49.

En la siguiente figura, se puede ver la trayectoria de una de las caminatas aleatorias que logro exitosamente volver a casa.

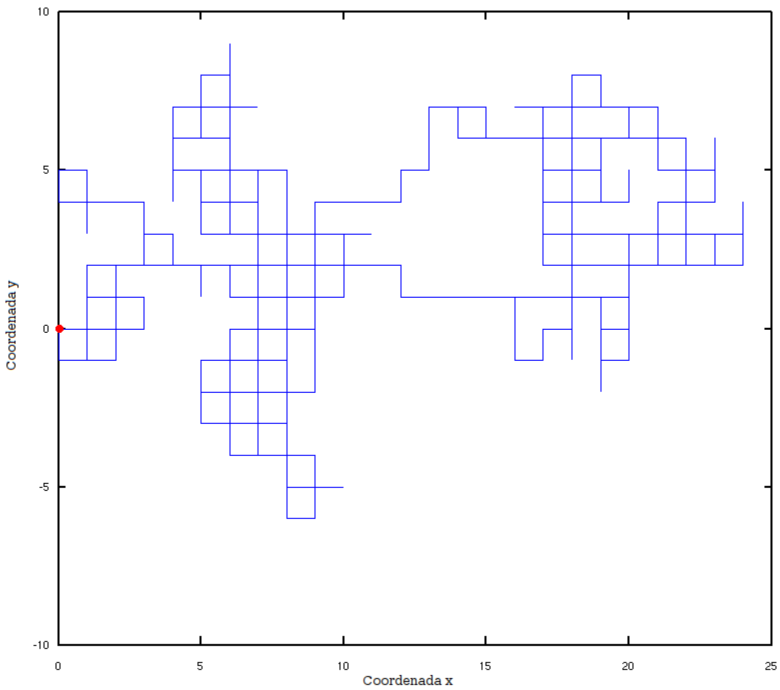


**Figura 3:** Trayectoria de una caminata aleatoria 2D

### Caminata aleatoria 3D

Para la caminata aleatoria 3D si consideramos un número de pasos igual a 10000 y de simulaciones igual a 10000, el código que utilizamos anteriormente para las caminatas aleatorias 1D y 2D, por estar escrito en Octave, tarda un tiempo no despreciable en terminar (aproximadamente entre 4 y 7 horas). Por esta razón, decidimos traducir el código al lenguaje C, que se caracteriza por ser muy eficiente. Aquí se debe hacer una

salvedad, ya que el la función $rand()$ de C no devuelve numeros aleatorios uniformemente disctribuidos. Para ver como resolvimos este inconveniente dirigirse al apéndice XXXXXXXX.
Hechas estas aclaraciones, los resultados arrojados fueron:

Probabilidad de la vuelta a casa: 0.3328
Promedio de pasos hasta la vuelta: 72.81

En la siguiente figura, se puede ver la trayectoria de una de las caminatas aleatorias que logro exitosamente volver a casa.

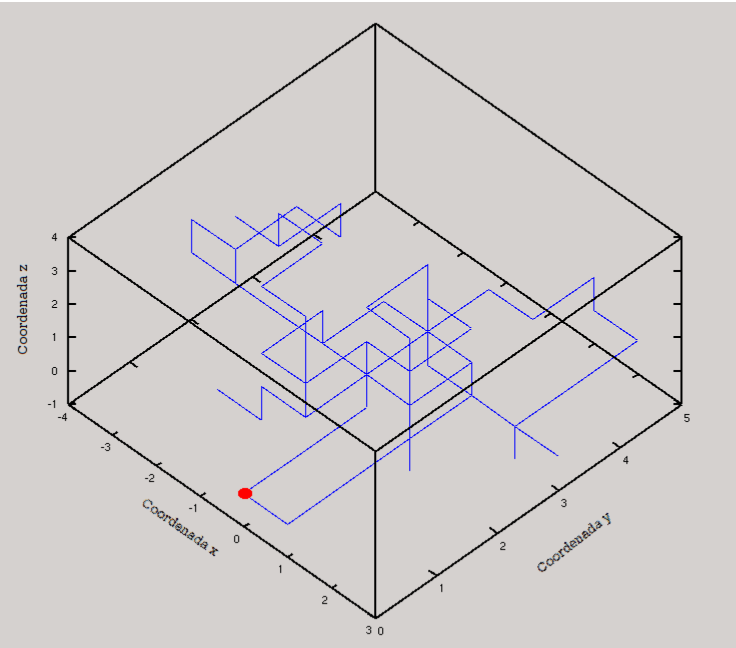


**Figura 4:** Trayectoria de una caminata aleatoria 3D

## 3. Apéndice: Código

Este es el código de Octave resuelve una caminata aleatoria de dimensión $dim$ con una cantidad máxima de pasos $maxPasos$.

```
function M = pasosBorracho(dim, maxPasos)

    M = zeros(dim, 1);
    llego = 0;

    for i = 1:maxPasos
        pos_anterior = M(:,end);
        direccion = floor(rand() * dim)+1;

        delta = 1;
        if rand() < 0.5
            delta = -1;
        endif
        pos_anterior(direccion) += delta;

        M = [M, pos_anterior];

        if pos_anterior == zeros(dim, 1)
            llego = 1;
            break;
        endif
    endfor

    if (llego == 0)
        M = zeros(dim, 1);
    endif
```

Este código genera las caminatas aleatorias para representarlas, teniendo en cuenta que sean lo suficientemente largas como para que se puede apreciar.

```
    recorrido1 = [];
    recorrido2 = [];
    recorrido3 = [];

    while size(recorrido1) < 30 || size(recorrido1) > 100
        recorrido1 = pasosBorracho(1, 10000);
    end

    while size(recorrido2) < 30 || size(recorrido2) > 100
        recorrido2 = pasosBorracho(2, 10000);
    end

    while size(recorrido3) < 30 || size(recorrido3) > 100
        recorrido3 = pasosBorracho(3, 10000);
    end

    plot(1:size(recorrido1)(2), recorrido1)
    plot(recorrido2(1,:), recorrido2(2,:))
    plot3(recorrido3(1,:), recorrido3(2,:), recorrido3(3,:))
```

Este es el código en C hace $ITERATIONS$ simulaciones de caminatas aleatorias de dimensión $dim$ con un máximo de pasos $maxPasos$.

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <time.h>

```

```c
#define MAXPASOS 10000
#define ITERATIONS 10000
#define DIM 1

int pasos[DIM][MAXPASOS];
int anterior[DIM];
bool llego;
int n;

inline int realrand(){
    if (DIM == 3) {
        int a = RAND_MAX;
        while (a >= RAND_MAX - 1) a = rand();
        return a;
    } else {
        return rand();
    }
}

void pasosBorracho(void) {
    llego = false;

    for(int i = 0; i < DIM; i++) {
        pasos[i][0] = 0;
    }

    for(n = 0; !llego && n < MAXPASOS; n++) {
        for(int i = 0; i < DIM; i++) {
            anterior[i] = pasos[i][n];
        }
        int direccion = realrand() %DIM;
        int delta = 1;
        if (realrand() % 2)
            delta = -1;

        anterior[direccion] += delta;

        for(int i = 0; i < DIM; i++)
            pasos[i][n+1] = anterior[i];

        bool yallego = true;
        for (int i = 0; yallego && i < DIM; i++)
            if (anterior[i] != 0)
                yallego = false;

        if (yallego)
            llego = true;
    }
}

int main() {

    srand(time(NULL));

    for(int i = 0; i < ITERATIONS; i++) {
        pasosBorracho();
        if (n != MAXPASOS) {
            for(int i = 0; i <= n; i++)  {
                for (int j = 0; j < DIM; j++) {
                    printf(" %d ", pasos[j][i]);
                }
                printf("\n");
```

```
            }
        } else {
            printf("0\n");
        }
        printf("\n");
    }
}
```

Este código en Python hace TEBEXXX COMPLETE!!!

```
if __name__ == '__main__':

    input = raw_input("Input file: ")
    output = raw_input("Output file: ")

    e = open(input, 'r').readlines()
    sizes = []
    n = 0

    for i in e:
        if i == '\n':
            if n != 0:
                sizes.append(n)
            n = 0
        else:
            n += 1

    f = open(output, 'w')

    for i in sizes:
        f.write('%d\n' %i)

    f.close()
```

Este fragmento de código en Octave hace las simulaciones y estima la probabilidad de la vuelta a casa y el promedio de pasos hasta la vuelta.

```
f = input("Filename: ", "s");
s = load(f);
v = []

for i=1:size(s)(1)
    if s(i) != 1
        v = [v, s(i)];
    end
end

printf("De 10000, s lo %d volvieron al inicio\n", size(v)(2));
printf("Promedio de pasos necesarios: %f\n", sum(v)/size(v)(2));

bar(hist(v, [1, 5, 25, 625, 10000]))
```