

10.2 Random Walk

File: *RandomWalk.nb*

Introduction to Computational Science: Modeling and Simulation for the Sciences
Angela B. Shiflet and George W. Shiflet
Wofford College
© 2006 by Princeton University Press

Show the path and distance of last position from first

```

(* determine points of path *)
SeedRandom[]
x = x0 = 0;
y = y0 = 0;
n = 25; (* number of steps *)
lst = {{0, 0}};
Do[
  If [RandomInteger[] == 0, x++, x--];
  If [RandomInteger[] == 0, y++, y--];
  lst = Append[lst, {x, y}],
  {n}]

(* display points in blue *)
lp1 = ListPlot[lst,
  AxesOrigin -> {0, 0},
  PlotStyle -> {PointSize[0.03], RGBColor[0, 0, 1]},
  DisplayFunction -> Identity];

(* plot lines connecting points *)
lp2 = ListPlot[lst,
  AxesOrigin -> {0, 0},
  Joined -> True, (* connect the dots *)
  DisplayFunction -> Identity];

(* display first and last points in larger in blue *)
lastPoint = ListPlot[{{0, 0}, {x, y}},
  AxesOrigin -> {0, 0},
  PlotStyle -> {PointSize[0.04], RGBColor[0, 0, 1]},
  DisplayFunction -> Identity];

(* show points and lines of path *)
Show[{lp1, lp2, lastPoint},
  DisplayFunction -> $DisplayFunction
]

(* distance between first and last points *)

$$\sqrt{(x - x0)^2 + (y - y0)^2} // N$$


```

Average over many runs to find average distance between first and last points

```
(* set up *)
SeedRandom[]
numTests = 100; (* number of runs *)
x0 = 0;
y0 = 0;
n = 25;          (* number of steps *)
sumDist = 0;    (* sum of distances *)

(* generate numTests number of random paths *)
Do[
  x = x0;
  y = y0;

  (* generate a random path *)
  Do[
    If [RandomInteger[] == 0, x++, x--];
    If [RandomInteger[] == 0, y++, y--],
    {n}];

  (* distance between first and last points *)
  sumDist +=  $\sqrt{(x - x0)^2 + (y - y0)^2}$ ,
  {numTests}];

(* average distance between first and last points *)
N[sumDist / numTests]
```

Generate list of average distances in n steps, where n varies from 1 to

50

```
(* set up *)
SeedRandom[]
numTests = 100; (* number of runs *)
x0 = 0;
y0 = 0;
Clear[n] (* n is number of steps *)
listDist = {}; (* list of distances *)

(* travel n steps for n from 1 to 50 *)
Do[

  (* generate numTests random paths of n steps each *)
  sumDist = 0;
  Do[
    x = x0;
    y = y0;

    (* generate a random path *)
    Do[
      If [RandomInteger[] == 0, x++, x--];
      If [RandomInteger[] == 0, y++, y--],
      {n}];

    (* distance between first and last points *)
    sumDist +=  $\sqrt{(x - x0)^2 + (y - y0)^2}$ ,
    {numTests}];

  (* append average distance traveled in n steps *)
  listDist = Append[listDist, N[sumDist / numTests]],
  {n, 50}]

(* return list of distances *)
listDist
```

Determine the relationship of distance to n , the number of steps by discovering an empirical model to fit the data.

```
lp = ListPlot[listDist, AxesLabel → {"n", "avg. dist."}]
```