# Apple-Bread-1.R

Admin

2024-03-16

```r
#Kangna Patel 2322
#Apple Bread Challenge 1

set.seed(2005)

#(i)
generate_pairs <- function(n){
  x <- numeric(n + 1)
  y <- numeric(n + 1)

  x[1] <- 0
  y[1] <- 0


  for (i in 2:(n + 1)) {
    direction <- sample(c(-1, 1), 1)
    # -1 means down or left and +1 means up or right
    if (runif(1) > 0.5) {
      x[i] <- x[i-1] + direction
      y[i] <- y[i-1]
    } else {
      x[i] <- x[i-1]
      y[i] <- y[i-1] + direction
    }
    #We chose whether to move vertically or horizontally, both cases having 50%
    #chances of moving
  }
  return(data.frame(x, y))
}


ordered_pairs <- generate_pairs(10)
print(ordered_pairs)
```

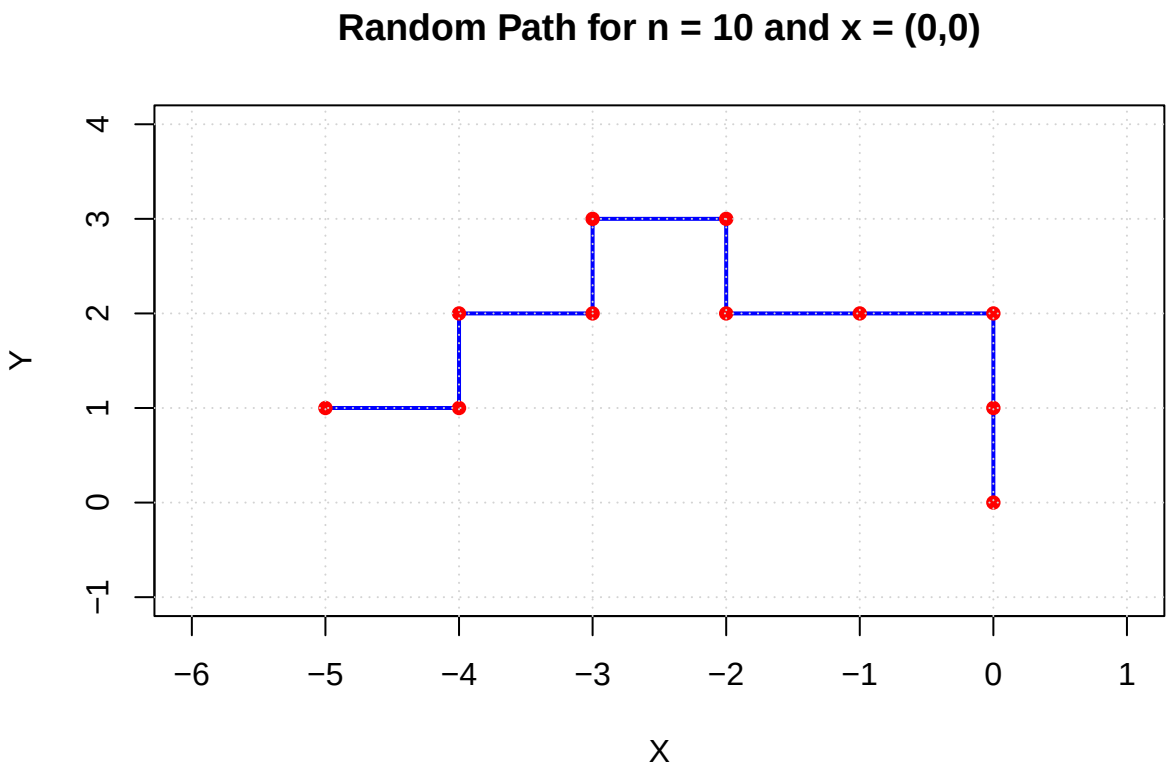
```
##     x y
## 1   0 0
## 2   0 1
## 3   0 2
## 4  -1 2
## 5  -2 2
## 6  -2 3
## 7  -3 3
```

```
## 8  -3 2
## 9  -4 2
## 10 -4 1
## 11 -5 1
```

```r
plot(ordered_pairs$x, ordered_pairs$y,
     type = "l",
     col = "blue",                      # Change line color to blue
     lwd = 2,                           # Line width
     lty = "solid",                     # Line type
     xlab = "X",
     ylab = "Y",
     main = "Random Path for n = 10 and x = (0,0)",
     xlim = c(min(ordered_pairs$x) - 1, max(ordered_pairs$x) + 1),
     ylim = c(min(ordered_pairs$y) - 1, max(ordered_pairs$y) + 1))

# Add red points
points(ordered_pairs$x, ordered_pairs$y, col = "red", pch = 16)

grid()
```




```r
#(ii)

ordered_pairs <- generate_pairs(100)
print(ordered_pairs)
```

```
##       x  y
## 1     0  0
## 2     1  0
## 3     1  1
## 4     1  0
## 5     1 -1
## 6     1 -2
## 7     2 -2
## 8     2 -3
## 9     3 -3
## 10    3 -2
## 11    2 -2
## 12    2 -1
## 13    3 -1
## 14    3  0
## 15    4  0
## 16    3  0
## 17    3 -1
## 18    3 -2
## 19    3 -1
## 20    4 -1
## 21    4  0
## 22    3  0
## 23    3  1
## 24    2  1
## 25    2  0
## 26    2 -1
## 27    3 -1
## 28    3  0
## 29    4  0
## 30    4  1
## 31    3  1
## 32    2  1
## 33    3  1
## 34    4  1
## 35    4  2
## 36    4  3
## 37    5  3
## 38    6  3
## 39    6  4
## 40    7  4
## 41    6  4
## 42    7  4
## 43    6  4
## 44    5  4
## 45    4  4
## 46    3  4
## 47    3  5
## 48    3  4
## 49    3  3
## 50    3  4
## 51    3  3
## 52    4  3
## 53    5  3
```

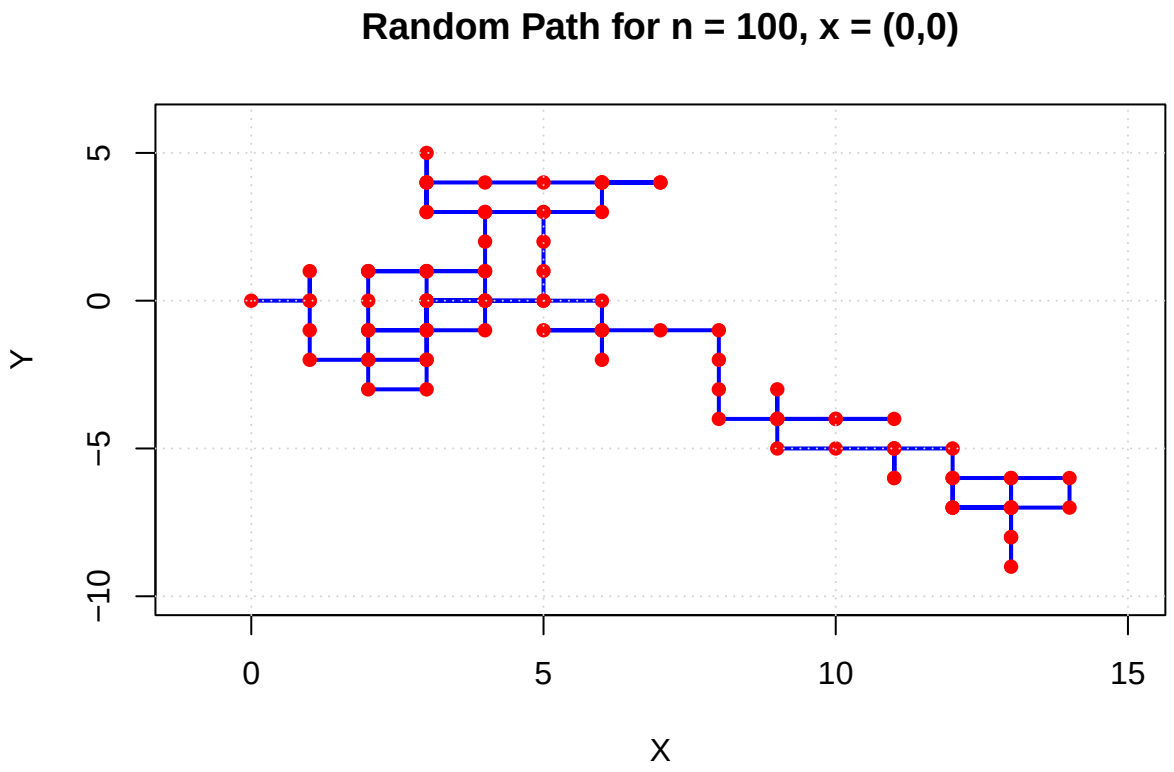
```
## 54   5  2
## 55   5  1
## 56   5  0
## 57   4  0
## 58   5  0
## 59   6  0
## 60   6 -1
## 61   6 -2
## 62   6 -1
## 63   5 -1
## 64   6 -1
## 65   7 -1
## 66   8 -1
## 67   8 -2
## 68   8 -3
## 69   8 -4
## 70   9 -4
## 71   9 -3
## 72   9 -4
## 73  10 -4
## 74  11 -4
## 75  10 -4
## 76   9 -4
## 77   9 -5
## 78  10 -5
## 79  11 -5
## 80  11 -6
## 81  11 -5
## 82  11 -6
## 83  11 -5
## 84  12 -5
## 85  12 -6
## 86  13 -6
## 87  14 -6
## 88  14 -7
## 89  13 -7
## 90  12 -7
## 91  13 -7
## 92  12 -7
## 93  12 -6
## 94  12 -7
## 95  13 -7
## 96  13 -8
## 97  13 -9
## 98  13 -8
## 99  13 -7
## 100 13 -6
## 101 13 -7
```

```r
plot(ordered_pairs$x, ordered_pairs$y,
     type = "l",
     col = "blue",                    # Change line color to blue
     lwd = 2,                         # Line width
     lty = "solid",                   # Line type
```

```
     xlab = "X",
     ylab = "Y",
     main = "Random Path for n = 100, x = (0,0)",
     xlim = c(min(ordered_pairs$x) - 1, max(ordered_pairs$x) + 1),
     ylim = c(min(ordered_pairs$y) - 1, max(ordered_pairs$y) + 1))

# Add red points
points(ordered_pairs$x, ordered_pairs$y, col = "red", pch = 16)

grid()
```




```
#(iii)
set.seed(2024)
#Considering taxicab distance :

u[1] <- 0
v[1] <- 0
i=2
distance[1] <- 0


while( distance[i-1] < 55){
    direction <- sample(c(-1, 1), 1)
    # -1 means down or left and +1 means up or right
    if (runif(1) > 0.5) {
```

```
      u[i] <- u[i-1] + direction
      v[i] <- v[i-1]
    } else {
      u[i] <- u[i-1]
      v[i] <- v[i-1] + direction
    }
    #We chose whether to move vertically or horizontally, both cases having 50%
    #chances of moving
    distance[i] <- abs(u[i]) + abs(v[i])
    i=i+1
  }

data <- data.frame(u,v)
head(data)
```

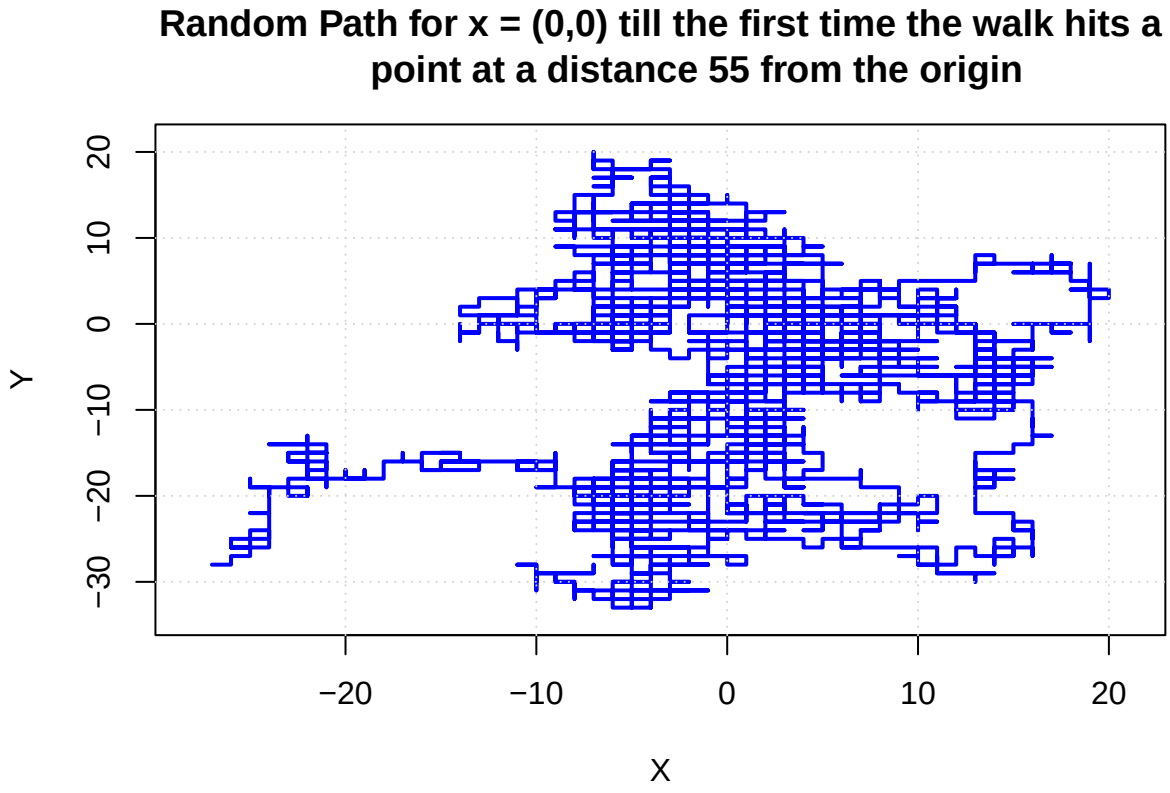
```
##    u  v
## 1  0  0
## 2  0  1
## 3 -1  1
## 4 -2  1
## 5 -2  0
## 6 -2 -1
```

```
tail(data)
```

```
##        u   v
## 3481 -26 -26
## 3482 -25 -26
## 3483 -25 -27
## 3484 -26 -27
## 3485 -26 -28
## 3486 -27 -28
```

```
plot(data$u, data$v,
       type = "l",
       col = "blue",                    # Change line color to blue
       lwd = 2,                         # Line width
       lty = "solid",                   # Line type
       xlab = "X",
       ylab = "Y",
       main = "Random Path for x = (0,0) till the first time the walk hits a
       point at a distance 55 from the origin",
       xlim = c(min(data$u) - 1, max(data$u) + 1),
       ylim = c(min(data$v) - 1, max(data$v) + 1))

grid()
```




```r
#(iv)
set.seed(4535210)
h <- numeric()
k <- numeric()
h[1] <- 55
k[1] <- 0
i=2

while(abs(h[i-1])>40 | abs(k[i-1])>40){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move vertically or horizontally, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>25 | abs(k[i-1])>25){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
```

```
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move vertically or horizontally, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>20 | abs(k[i-1])>20){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>17 | abs(k[i-1])>17){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>15 | abs(k[i-1])>15){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
```

```
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>13 | abs(k[i-1])>13){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>11 | abs(k[i-1])>11){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>10 | abs(k[i-1])>10){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>7 | abs(k[i-1])>7){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
```

```
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>5 | abs(k[i-1])>5){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>2 | abs(k[i-1])>2){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
  i=i+1
}

while(abs(h[i-1])>1 | abs(k[i-1])>1){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move verticallk or horizontallk, both cases having 50%
  #chances of moving
```

```
  i=i+1
}

while(h[i-1]!=0 | k[i-1] !=0 ){
  direction <- sample(c(-1, 1), 1)
  # -1 means down or left and +1 means up or right
  if (runif(1) > 0.5) {
    h[i] <- h[i-1] + direction
    k[i] <- k[i-1]
  } else {
    h[i] <- h[i-1]
    k[i] <- k[i-1] + direction
  }
  #We chose whether to move vertically or horizontally, both cases having 50%
  #chances of moving
  i=i+1
}


data2 <- data.frame(h,k)
head(data2)
```

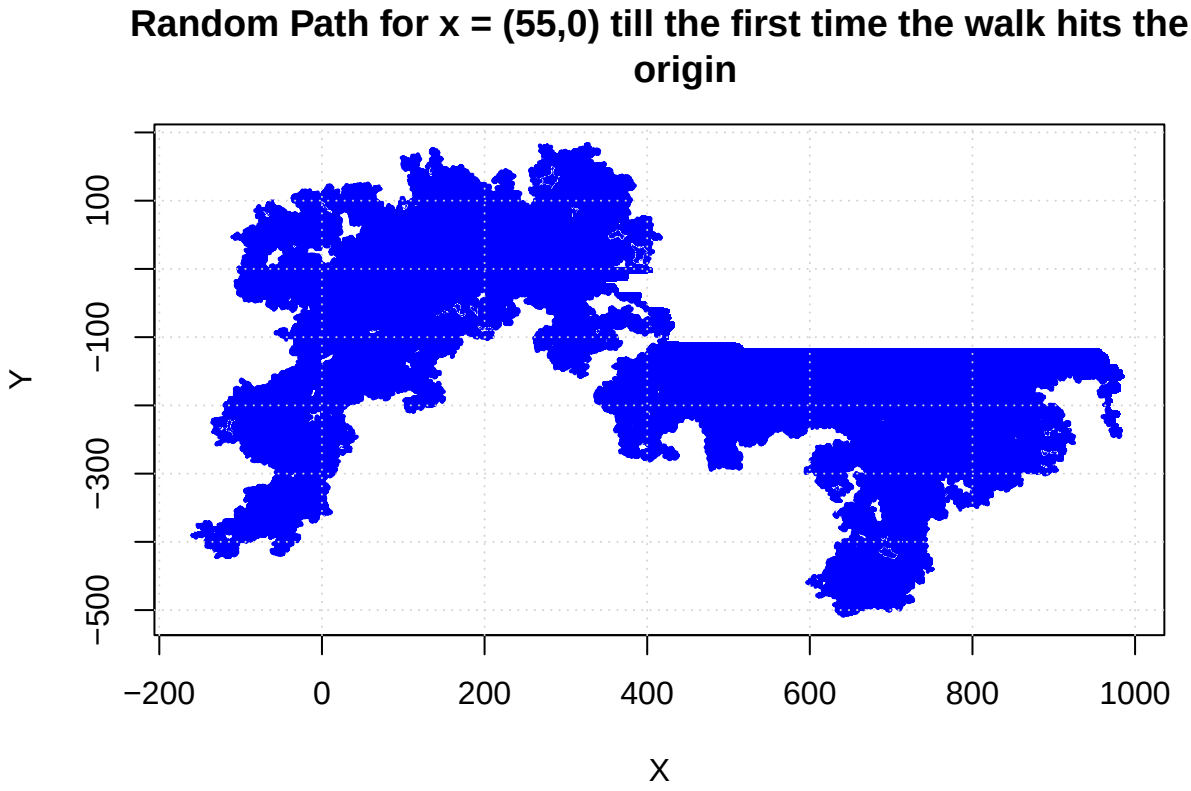
```
##    h k
## 1 55 0
## 2 56 0
## 3 56 1
## 4 55 1
## 5 54 1
## 6 53 1
```

```
tail(data2)
```

```
##          h k
## 1159561  0 1
## 1159562  0 2
## 1159563  0 1
## 1159564 -1 1
## 1159565  0 1
## 1159566  0 0
```

```
plot(data2$h, data2$k,
     type = "l",
     col = "blue",                    # Change line color to blue
     lwd = 2,                         # Line width
     lty = "solid",                   # Line type
     xlab = "X",
     ylab = "Y",
     main = "Random Path for x = (55,0) till the first time the walk hits the
     origin",
     xlim = c(min(data2$h) - 1, max(data2$h) + 1),
     ylim = c(min(data2$k) - 1, max(data2$k) + 1))

grid()
```


Random Path for x = (55,0) till the first time the walk hits the origin
Y
100
−100
−300
−500
−200
0
200
400
600
800
1000
X