Practice Final Question Solutions

Cindy Rush (cgr2130)

Section

Instructions (Read this completely first)

You should complete the exam by editting this file directly. Please knit the file often, so that if you make a mistake you catch it before the end of the exam. You will have exactly 20 minutes from the start time to complete the exam. **At the end you must turn in your knitted .pdf file and raw .Rmd file on Courseworks.**

When the time is up, you must shut your computer immediately. We will take off points from anyone whose computer is still open after time is up.

You may use your class notes for the exam, but not the internet. You absolutely may not communicate with anyone else during the exam. Doing so will result in an F in this class and likely result in termination from the MA program. Note that your time will be tight so you will not be able to look up every bit of code from your class notes.

Question 1 (30 points)

Consider the following "random walk" procedure:

1. Start with $x = 5$
2. Draw a random number r uniformly between -2 and 1.
3. Replace x with $x + r$
4. Stop if $x \leq 0$
5. Else repeat

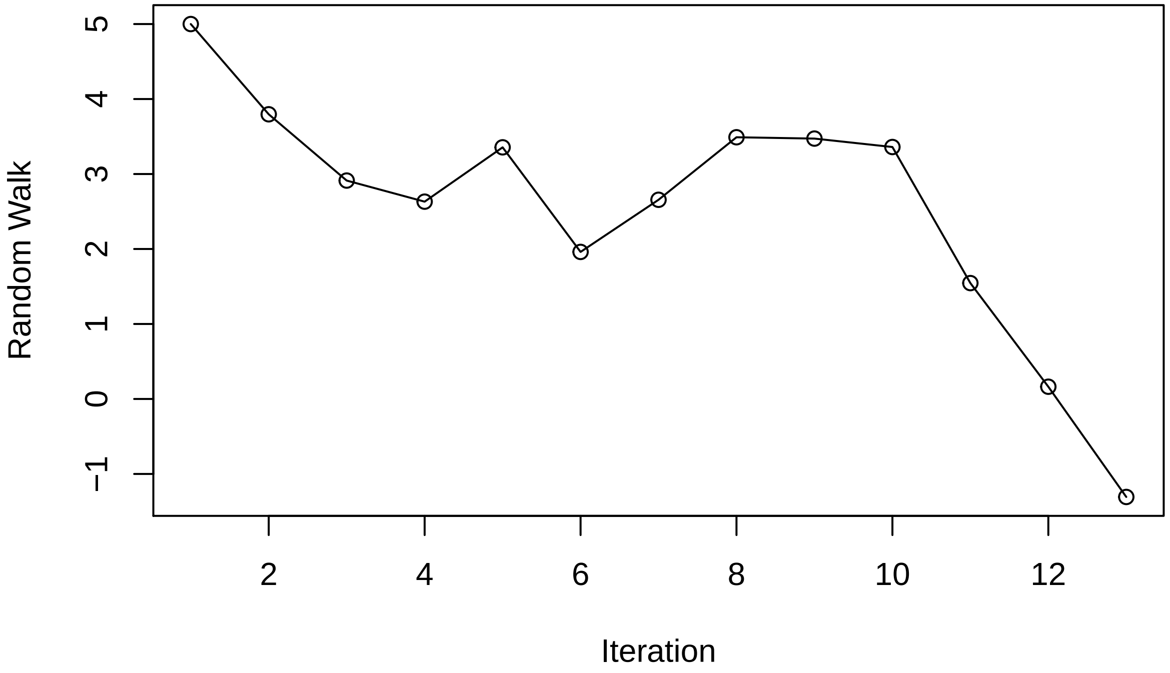
a. Write a `while()` loop to implement this procedure. Importantly, save all the positive values of x that were visited in this procedure in a vector called `x.vals`, and display its entries.

```
set.seed(1)

x.vals <- 5
i <- 1
while(x.vals[i] > 0) {
  r      <- runif(1, min = -2, max = 1)
  x.vals <- c(x.vals, x.vals[i] + r)
  i      <- i+1
}
x.vals
```

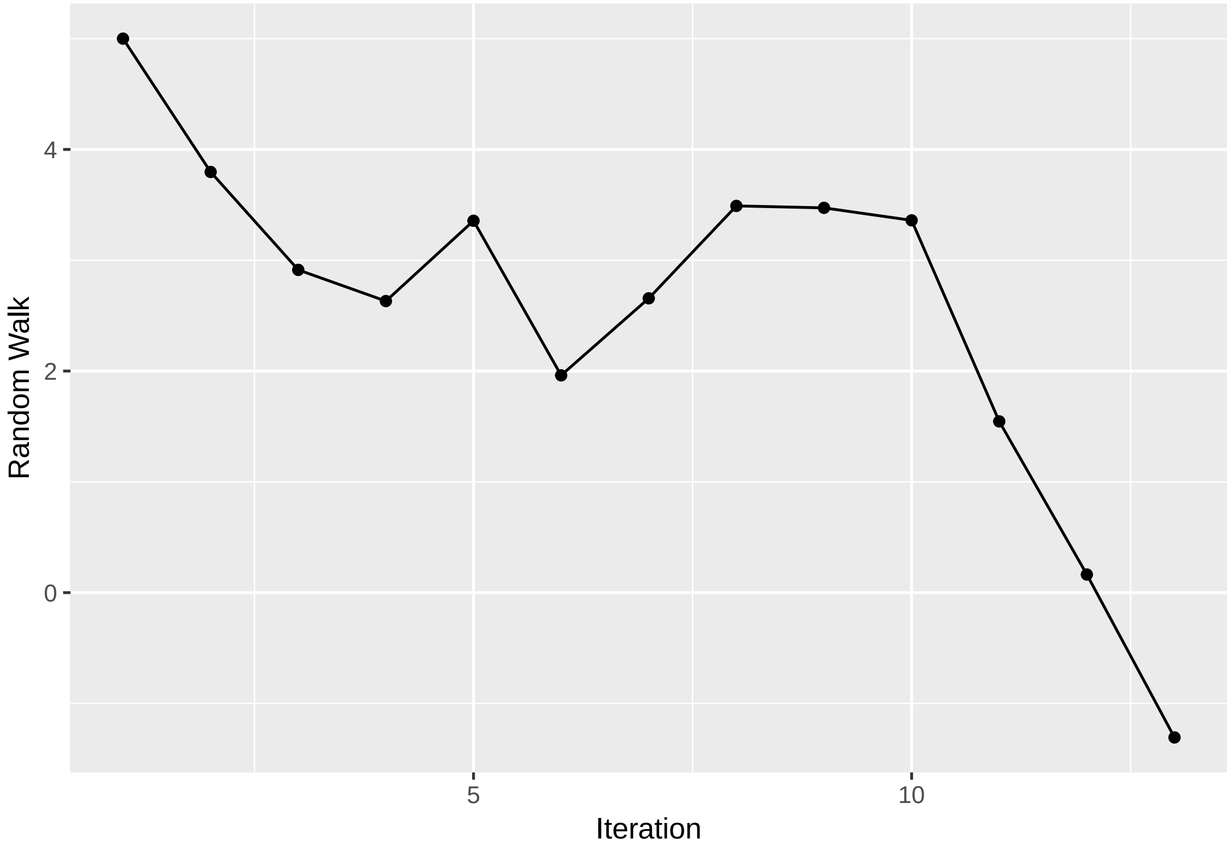
```
##  [1]  5.0000000  3.7965260  2.9128977  2.6314578  3.3560811  1.9611269
##  [7]  2.6562960  3.4903218  3.4727152  3.3600573  1.5454161  0.1633398
## [13] -1.3069899
```

b. Produce a plot of the random walk values `x.vals` from above versus the iteration number. Make sure the plot has an appropriately labeled x-axis and and y-axis. You should plot points for the location of the random walk at each iteration and connect the points with a line.

```
plot(1:length(x.vals), x.vals, type = "o", xlab = "Iteration", ylab = "Random Walk")

library(ggplot2)
```



```
walk <- data.frame(iteration = 1:length(x.vals), value = x.vals)

ggplot(walk) +
  geom_line(mapping = aes(x = iteration, y = value)) +
  geom_point(mapping = aes(x = iteration, y = value)) +
  labs(x = "Iteration", y = "Random Walk")
```



c. (8 points) Make sure you have completed the above before you begin part c. Below is a `random.walk()` function made to implement the random walk procedure. Unfortunately, the function has some bugs – find them and fix them!

```
random.walk <- function(x.start = 5, seed = NULL) {
  if (!is.null(seed)) set.seed(seed)
  x.vals <- x.start
  while (TRUE) {
    r <- runif(1, -2, 1)
    if (tail(x.vals + r, 1) <= 0) {
      break
    } else {
      x.vals <- c(x.vals, x.vals + r)
    }
  }
  return(x.vals = x.vals, num.steps = length(x.vals))
}

# random.walk(x.start = 5, seed = 3)$num.steps # Should print 8
# random.walk(x.start = 10, seed = 7)$num.steps # Should print 14
```

```
random.walk <- function(x.start = 5, seed = NULL) {
  if (!is.null(seed)) set.seed(seed)
  x.vals <- x.start
  while (TRUE) {
    r <- runif(1, -2, 1)
    if (tail(x.vals + r, 1) <= 0) {
      break
    } else {
      x.vals <- c(x.vals, x.vals[length(x.vals)] + r)
    }
  }
  return(list(x.vals = x.vals, num.steps = length(x.vals)))
}
random.walk(x.start = 5, seed = 3)$num.steps # Should print 8
```

```
## [1] 8
```

```
random.walk(x.start = 10, seed = 7)$num.steps # Should print 14
```

```
## [1] 14
```