# Semicircle problem with simulation

Zehui Yin

2022-10-15

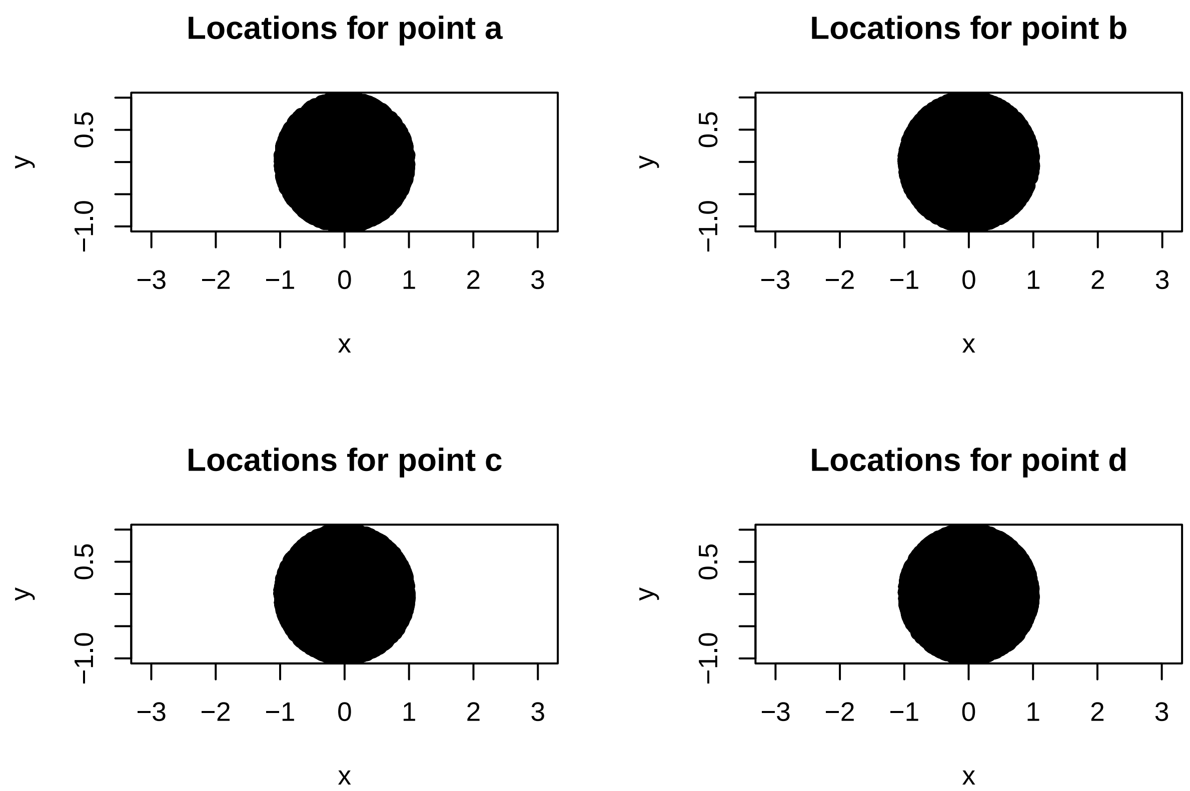
If we have four ducks swimming in a circle and their locations are random and independent, what is the probability that all four of the ducks are in the same half of the circle? In this project, I solved this problem through simulation in R. I found that the probability of this to happen is around 50%.

We first generate the duck locations

```r
# create a function to genereate random location in a circle
generate_point_in_circle <- function(n, radius){
  output <- data.frame()
  while (nrow(output) < n) {
    iteration <- runif(2, min = -radius, max = radius)
    if (iteration[1]^2 + iteration[2]^2 <= radius^2) {
      output <- rbind(output, iteration)
    }
  }
  colnames(output) <- c("x", "y")
  return(output)
}

# generate 10000 samples of set of four points
n <- 10000
a <- generate_point_in_circle(n, 1)
b <- generate_point_in_circle(n, 1)
c <- generate_point_in_circle(n, 1)
d <- generate_point_in_circle(n, 1)

par(mfrow = c(2,2))
plot(a[, "x"], a[, "y"], asp=1,
     main = "Locations for point a", xlab = "x", ylab = "y")
plot(b[, "x"], b[, "y"], asp=1,
     main = "Locations for point b", xlab = "x", ylab = "y")
plot(c[, "x"], c[, "y"], asp=1,
     main = "Locations for point c", xlab = "x", ylab = "y")
plot(d[, "x"], d[, "y"], asp=1,
     main = "Locations for point d", xlab = "x", ylab = "y")
```




Then we check for each set of points whether they fall within the same semicircle or not.

```
# create variable to record the number of TRUE happens
count_ture <- 0
for (i in 1:n) {
  cond <- 0
  # check whether b, c, d falls within the semicircle created by a and origin (0,0)
  if (b[i,"y"]*a[i,"x"]-b[i,"x"]*a[i,"y"] > 0 &
      c[i,"y"]*a[i,"x"]-c[i,"x"]*a[i,"y"] > 0 &
      d[i,"y"]*a[i,"x"]-d[i,"x"]*a[i,"y"] > 0) {
    cond <- cond + 1
  }
  if (b[i,"y"]*a[i,"x"]-b[i,"x"]*a[i,"y"] < 0 &
      c[i,"y"]*a[i,"x"]-c[i,"x"]*a[i,"y"] < 0 &
      d[i,"y"]*a[i,"x"]-d[i,"x"]*a[i,"y"] < 0) {
    cond <- cond + 1
  }
  # check whether a, c, d falls within the semicircle created by b and origin (0,0)
  if (a[i,"y"]*b[i,"x"]-a[i,"x"]*b[i,"y"] > 0 &
      c[i,"y"]*b[i,"x"]-c[i,"x"]*b[i,"y"] > 0 &
      d[i,"y"]*b[i,"x"]-d[i,"x"]*b[i,"y"] > 0) {
    cond <- cond + 1
  }
  if (a[i,"y"]*b[i,"x"]-a[i,"x"]*b[i,"y"] < 0 &
      c[i,"y"]*b[i,"x"]-c[i,"x"]*b[i,"y"] < 0 &
      d[i,"y"]*b[i,"x"]-d[i,"x"]*b[i,"y"] < 0) {
    cond <- cond + 1
```

```r
  }
  # check whether a, b, d falls within the semicircle created by c and origin (0,0)
  if (a[i,"y"]*c[i,"x"]-a[i,"x"]*c[i,"y"] > 0 &
      b[i,"y"]*c[i,"x"]-b[i,"x"]*c[i,"y"] > 0 &
      d[i,"y"]*c[i,"x"]-d[i,"x"]*c[i,"y"] > 0) {
    cond <- cond + 1
  }
  if (a[i,"y"]*c[i,"x"]-a[i,"x"]*c[i,"y"] < 0 &
      b[i,"y"]*c[i,"x"]-b[i,"x"]*c[i,"y"] < 0 &
      d[i,"y"]*c[i,"x"]-d[i,"x"]*c[i,"y"] < 0) {
    cond <- cond + 1
  }
  # check whether a, b, c falls within the semicircle created by d and origin (0,0)
  if (a[i,"y"]*d[i,"x"]-a[i,"x"]*d[i,"y"] > 0 &
      b[i,"y"]*d[i,"x"]-b[i,"x"]*d[i,"y"] > 0 &
      c[i,"y"]*d[i,"x"]-c[i,"x"]*d[i,"y"] > 0) {
    cond <- cond + 1
  }
  if (a[i,"y"]*d[i,"x"]-a[i,"x"]*d[i,"y"] < 0 &
      b[i,"y"]*d[i,"x"]-b[i,"x"]*d[i,"y"] < 0 &
      c[i,"y"]*d[i,"x"]-c[i,"x"]*d[i,"y"] < 0) {
    cond <- cond + 1
  }
  # if any of the 4 conditions is met, then the four points are in the same semicircle
  if (cond >= 1) {
    count_ture <- count_ture + 1
  }
}

# calculate the proportion
cat("Proportion of four ducks fall within the same semicircle \nwith", n,
    "simulations equals =", count_ture*100/n, "%")
```

```
## Proportion of four ducks fall within the same semicircle
## with 10000 simulations equals = 49.76 %
```

Based on the simulation we can see that the proportion is 49.76% when we have a large sample size. Then I want to see the changing trend in the average proportion with iteration increases. I first create a function to replicate the process above.

```r
ducks_simulation <- function(n) {
  a <- generate_point_in_circle(n, 1)
  b <- generate_point_in_circle(n, 1)
  c <- generate_point_in_circle(n, 1)
  d <- generate_point_in_circle(n, 1)
  count_ture <- 0
  for (i in 1:n) {
    cond <- 0
    # check whether b, c, d falls within the semicircle created by a and origin (0,0)
    if (b[i,"y"]*a[i,"x"]-b[i,"x"]*a[i,"y"] > 0 &
        c[i,"y"]*a[i,"x"]-c[i,"x"]*a[i,"y"] > 0 &
        d[i,"y"]*a[i,"x"]-d[i,"x"]*a[i,"y"] > 0) {
      cond <- cond + 1
    }
    if (b[i,"y"]*a[i,"x"]-b[i,"x"]*a[i,"y"] < 0 &
```

```r
        c[i,"y"]*a[i,"x"]-c[i,"x"]*a[i,"y"] < 0 &
        d[i,"y"]*a[i,"x"]-d[i,"x"]*a[i,"y"] < 0) {
      cond <- cond + 1
    }
    # check whether a, c, d falls within the semicircle created by b and origin (0,0)
    if (a[i,"y"]*b[i,"x"]-a[i,"x"]*b[i,"y"] > 0 &
        c[i,"y"]*b[i,"x"]-c[i,"x"]*b[i,"y"] > 0 &
        d[i,"y"]*b[i,"x"]-d[i,"x"]*b[i,"y"] > 0) {
      cond <- cond + 1
    }
    if (a[i,"y"]*b[i,"x"]-a[i,"x"]*b[i,"y"] < 0 &
        c[i,"y"]*b[i,"x"]-c[i,"x"]*b[i,"y"] < 0 &
        d[i,"y"]*b[i,"x"]-d[i,"x"]*b[i,"y"] < 0) {
      cond <- cond + 1
    }
    # check whether a, b, d falls within the semicircle created by c and origin (0,0)
    if (a[i,"y"]*c[i,"x"]-a[i,"x"]*c[i,"y"] > 0 &
        b[i,"y"]*c[i,"x"]-b[i,"x"]*c[i,"y"] > 0 &
        d[i,"y"]*c[i,"x"]-d[i,"x"]*c[i,"y"] > 0) {
      cond <- cond + 1
    }
    if (a[i,"y"]*c[i,"x"]-a[i,"x"]*c[i,"y"] < 0 &
        b[i,"y"]*c[i,"x"]-b[i,"x"]*c[i,"y"] < 0 &
        d[i,"y"]*c[i,"x"]-d[i,"x"]*c[i,"y"] < 0) {
      cond <- cond + 1
    }
    # check whether a, b, c falls within the semicircle created by d and origin (0,0)
    if (a[i,"y"]*d[i,"x"]-a[i,"x"]*d[i,"y"] > 0 &
        b[i,"y"]*d[i,"x"]-b[i,"x"]*d[i,"y"] > 0 &
        c[i,"y"]*d[i,"x"]-c[i,"x"]*d[i,"y"] > 0) {
      cond <- cond + 1
    }
    if (a[i,"y"]*d[i,"x"]-a[i,"x"]*d[i,"y"] < 0 &
        b[i,"y"]*d[i,"x"]-b[i,"x"]*d[i,"y"] < 0 &
        c[i,"y"]*d[i,"x"]-c[i,"x"]*d[i,"y"] < 0) {
      cond <- cond + 1
    }
    # if any of the 4 conditions is met, then the four points are in the same semicircle
    if (cond >= 1) {
      count_ture <- count_ture + 1
    }
  }
  return(count_ture*100/n)
}
```

Use parallel package to do parallel computing with the function. Like above, still simulate for 10000 times.

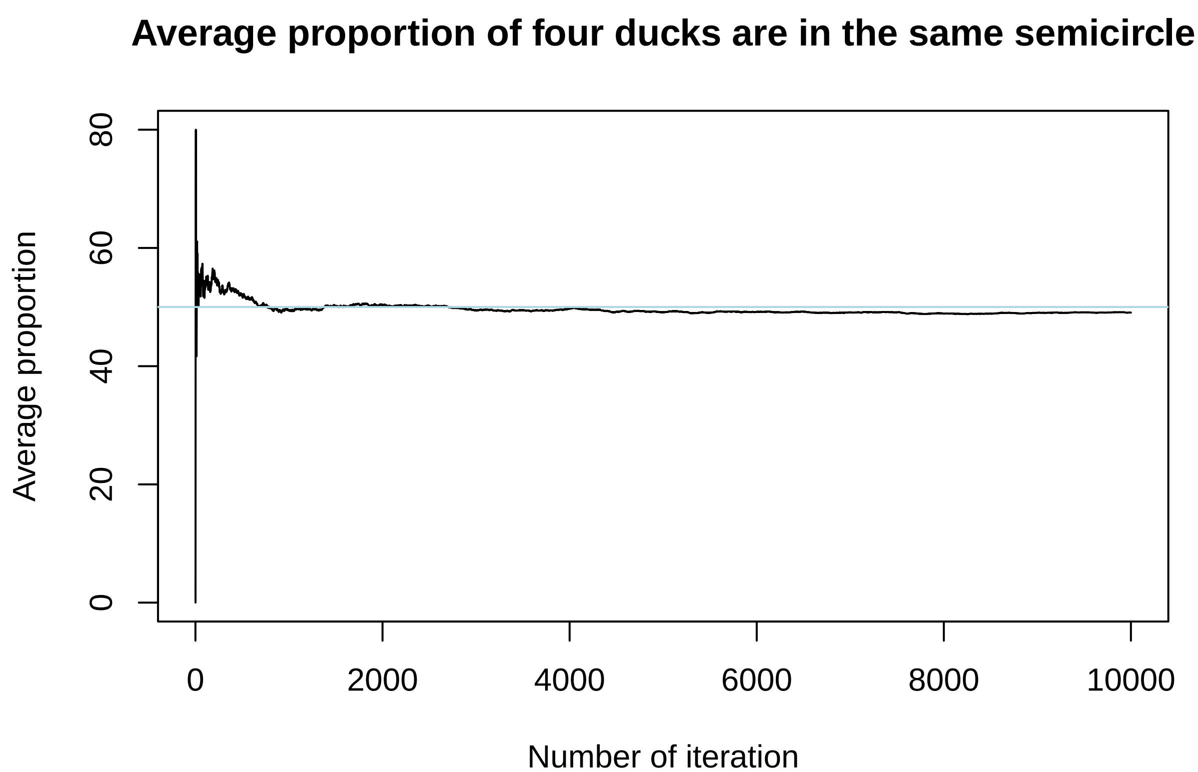
```r
library(parallel)
simulation_output <- mclapply(rep(1, 10000), ducks_simulation)
simulation_output <- unlist(simulation_output)
```

Visualize the average proportion trend with the increase of simulation times.

```r
pop_vis <- data.frame(matrix(nrow = 10000, ncol = 2))
colnames(pop_vis) <- c("first_n_iteration", "average proportion")
```

```r
for (i in 1:10000) {
  pop_vis[i,] <- c(i, mean(simulation_output[1:i]))
}

plot(pop_vis[, "first_n_iteration"], pop_vis[, "average proportion"],
     main = "Average proportion of four ducks are in the same semicircle",
     xlab = "Number of iteration", ylab = "Average proportion", type = "l",
     xlim = c(0, 10000),
     ylim = c(min(pop_vis[, "average proportion"]), max(pop_vis[, "average proportion"])))
abline(h = 50, col="lightblue")
```




The average proportion seems to quickly converge to 50%.