4b

Yiqiao Yin

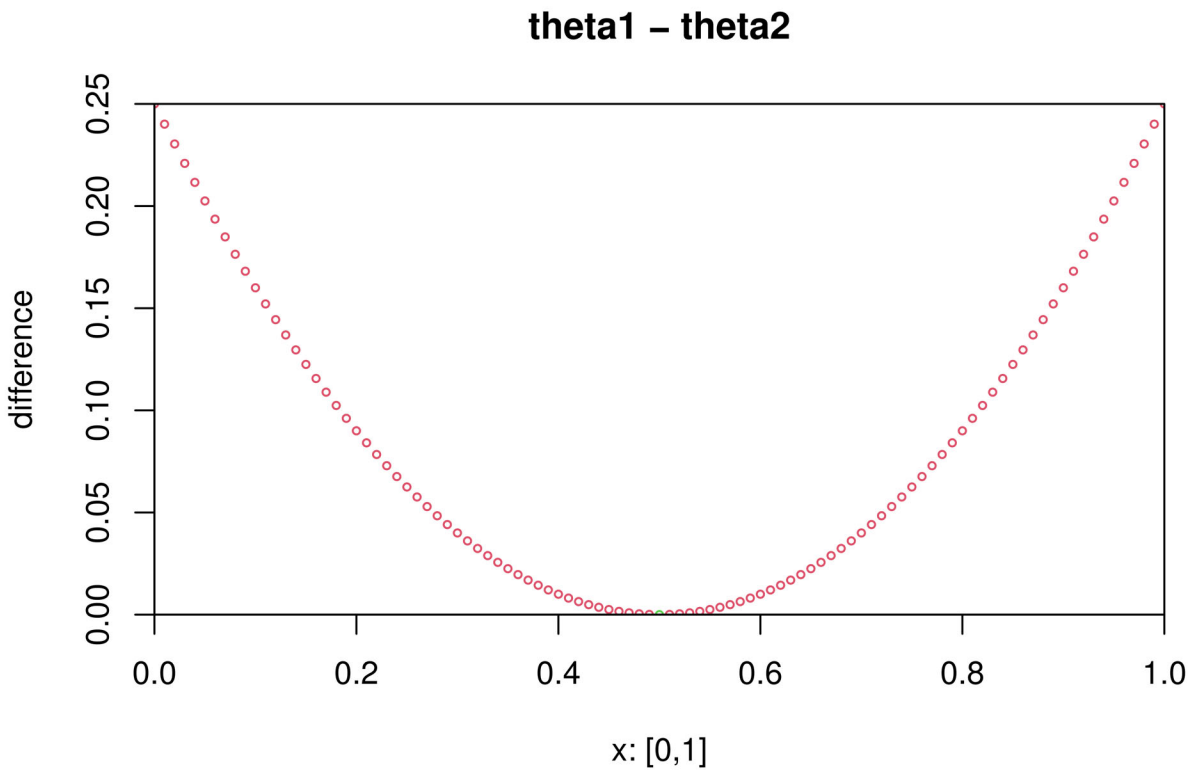
2/2/2021

Problem 1:

Please see the previous part (written) part attached to this.

I present the last part, i.e. the comparison of two estimates, within the restricted region of theta $[0, 1]$ in the following.

```
theta = seq(0, 1, 0.01)
plot(theta, theta^2 - theta + 1/3 - 1/12, col = as.numeric(theta^2 - theta + 1/3 - 1/12 < 0) + 2,
     type = "p", cex = 0.5,
     xaxs = "i", yaxs = "i",
     xlab = "x: [0,1]", ylab = "difference",
     main = "theta1 - theta2")
```



Problem 2:

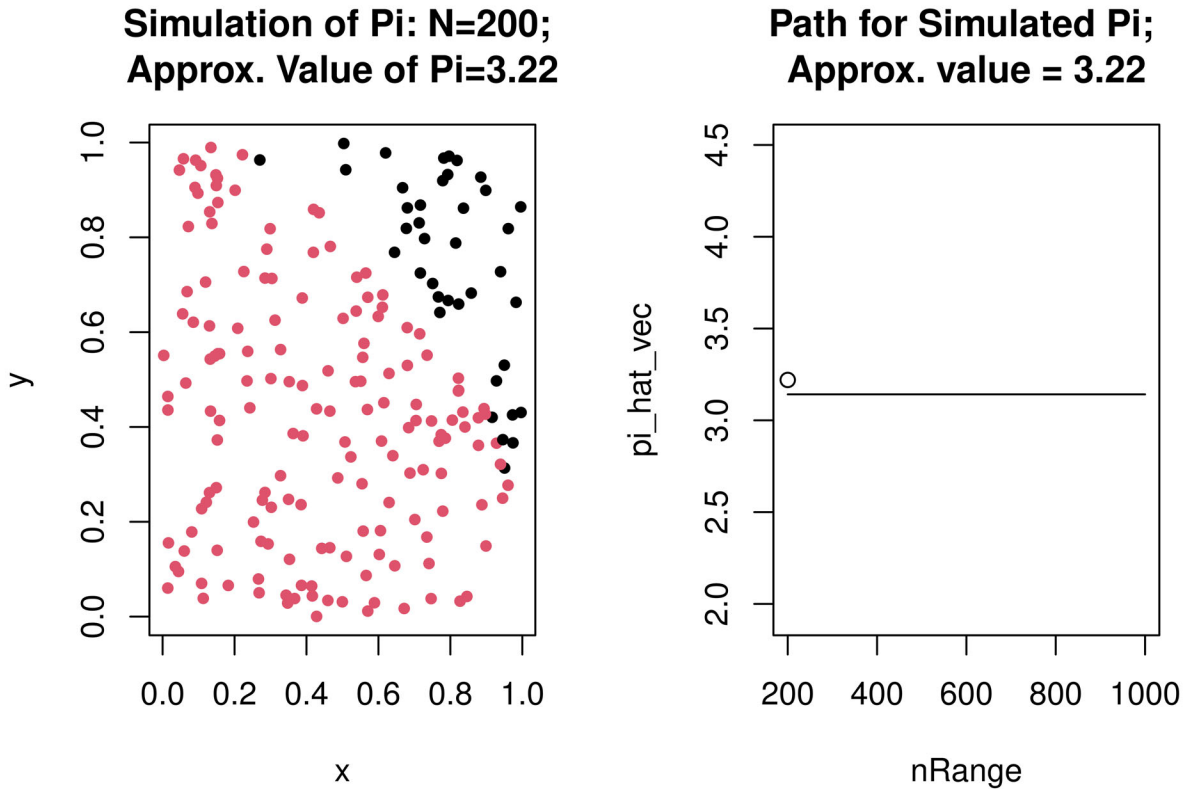
Consider a unit square. In other words, consider $X_1, X_2 \sim \text{uniform}([0,1])$ while X_1 and X_2 are independent. Then consider the following sequence with running index n:

$$y := \begin{cases} 4/n & \text{if } (x_1^2 + x_2^2)^2 \leq 1 \\ 0 & \text{else} \end{cases}$$

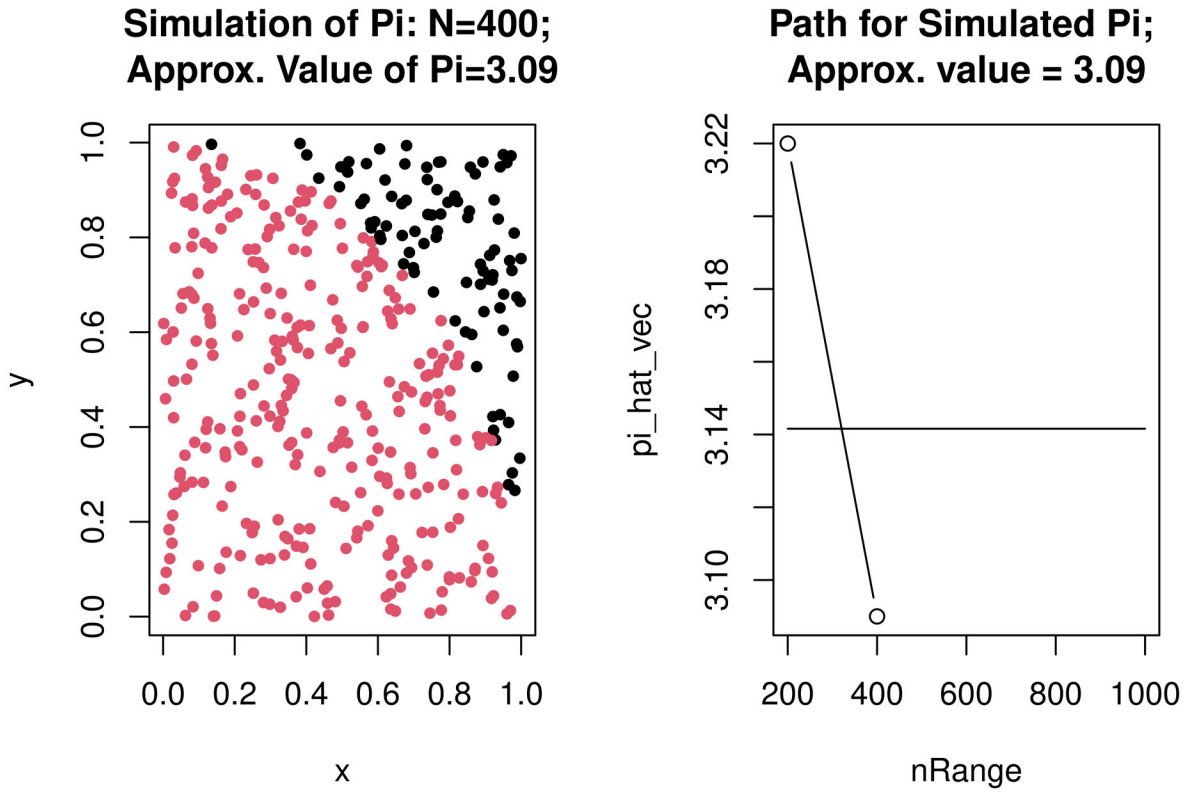
Let us simulate a unit square. Let us mark the dots (X_1, X_2) that with distance less than or equal to 1 in one color and the rest of the dots in another color. We count the dots to be estimated area of a quarter of a circle. Then we recover estimated π as $n \to \infty$.

```
## Define a function to output a plot of pi
nRange <- seq(0, 1e3, 2e2)[-1]
pi_hat_vec <- rep(NA, length(nRange))
for (N in nRange) {
  x <- runif(N)
  y <- runif(N)
  d <- sqrt(x^2 + y^2)
  label <- ifelse(d < 1, 1, 0)
  pi_hat <- round(4*plyr::count(label)[2,2]/N,3)
  pi_hat_vec[which(N == nRange)] <- pi_hat
  par(mfrow=c(1,2))
  plot(
    x, y,
    col = label+1,
    main = paste0(
      "Simulation of Pi: N=", N,
      "; \nApprox. Value of Pi=", pi_hat),
    pch = 20, cex = 1)
  plot(
    nRange, pi_hat_vec, type = "both",
    main = paste0("Path for Simulated Pi; \nApprox. value = ", pi_hat));
  lines(nRange, y = rep(pi, length(nRange)))
}
```

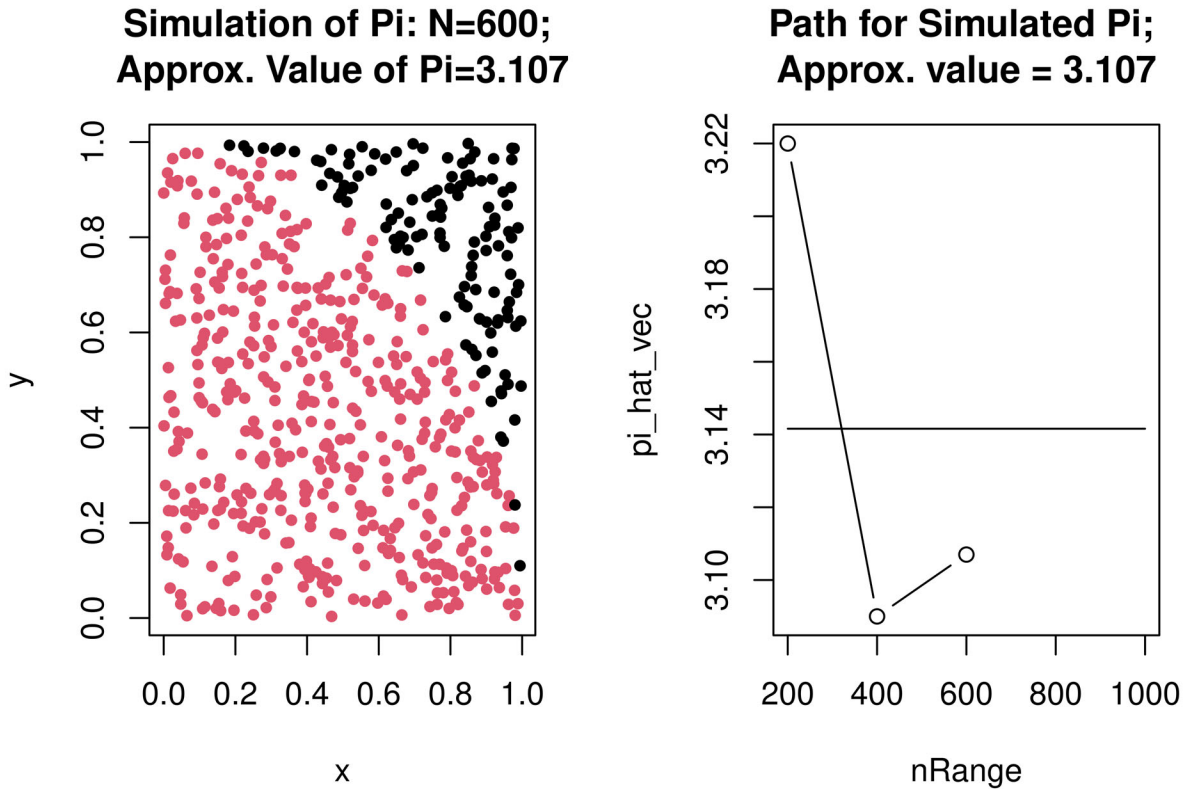
```
## Warning in plot.xy(xy, type, ...): plot type 'both' will be truncated to first
## character
```



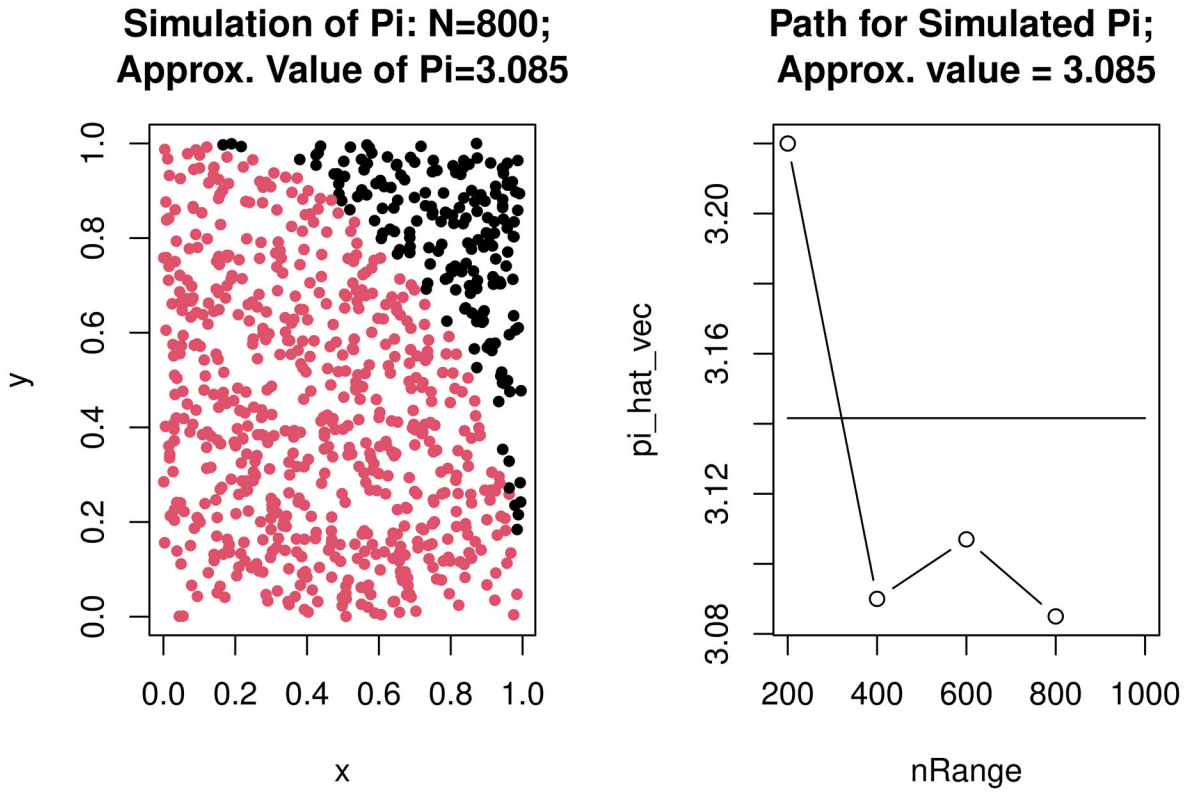

```
## Warning in plot.xy(xy, type, ...): plot type 'both' will be truncated to first
## character
```



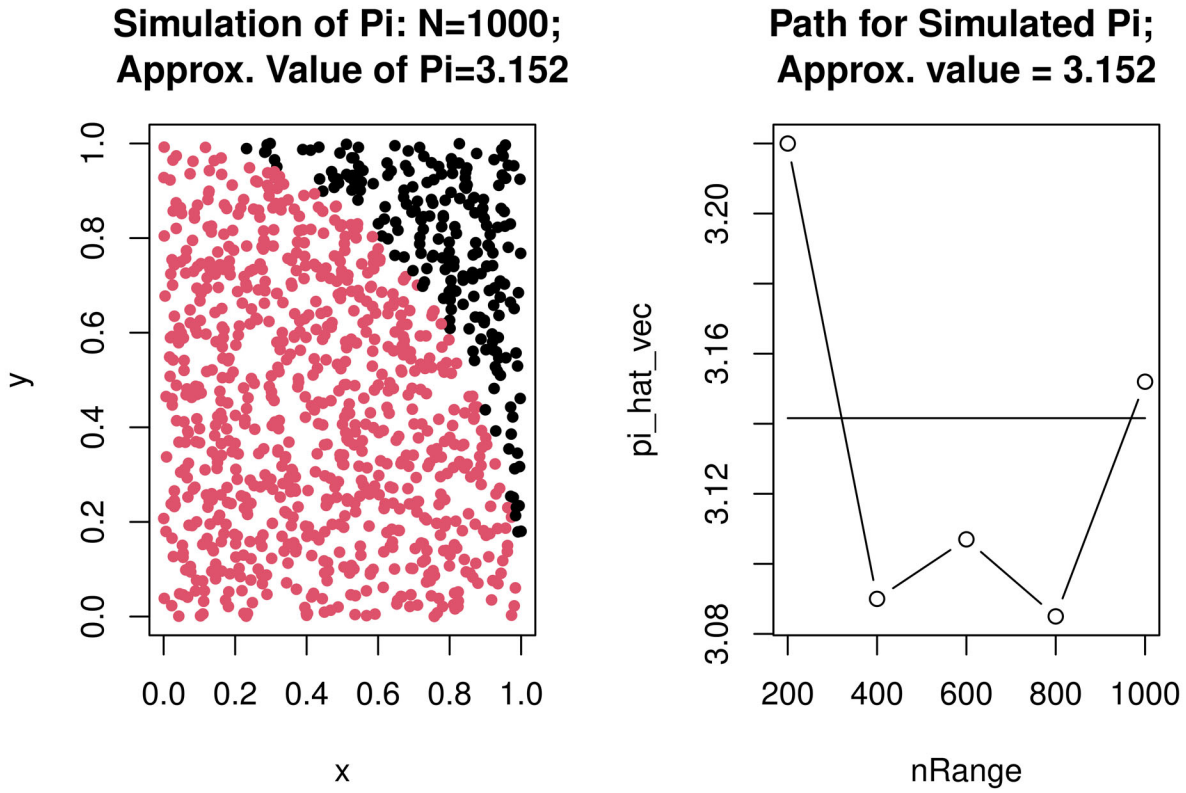

```
## Warning in plot.xy(xy, type, ...): plot type 'both' will be truncated to first
## character
```




```
## Warning in plot.xy(xy, type, ...): plot type 'both' will be truncated to first
## character
```




```
## Warning in plot.xy(xy, type, ...): plot type 'both' will be truncated to first
## character
```



Appendix:

Let me create the above simulation in more refined sequence of n's. In the following, I will create a small animation, of which I will save as "GIF" format and upload as a separate file.

```
# # Library
# library(animation)
#
# ## Plot Monte Carlo Simulation of Pi
# saveGIF({
#   ## Define a function to output a plot of pi
#   nRange <- seq(1e2, 1e4, 1e2)
#   pi_hat_vec <- rep(NA, length(nRange))
#   for (N in nRange) {
#     x <- runif(N)
#     y <- runif(N)
#     d <- sqrt(x^2 + y^2)
#     label <- ifelse(d < 1, 1, 0)
#     pi_hat <- round(4*plyr::count(label)[2,2]/N,3)
#     pi_hat_vec[which(N == nRange)] <- pi_hat
#     par(mfrow=c(1,2))
#     plot(
#       x, y,
#       col = label+1,
#       main = paste0(
#         "Simulation of Pi: N=", N,
#         "; \nApprox. Value of Pi=", pi_hat),
```

```
#       pch = 20, cex = 1)
#     plot(
#       nRange, pi_hat_vec, type = "both",
#       main = paste0("Path for Simulated Pi; \nApprox. value = ", pi_hat));
#     lines(nRange, y = rep(pi, length(nRange)))
#   }
# }, movie.name = "C:/Users/eagle/OneDrive/Course/CU Stats/STATS GR6102 - Applied Statistics II/InClass
# ani.width = 480)
```