

This computing project centers on different methods used to estimate the total of some attribute of a population that is distributed across a spatial region. To make this concrete, shown in Figure 1 is a small forest that contains a total of 3396 trees. The area of each circle is proportional to the total basal area of the tree, which is the cross-sectional area of the trunk 1.6 meters from the ground.

The objective of this project is to compare several different methods for estimating the total basal area of the 3396 trees in this forest, in order to compare their relative accuracy. For each method, you will write code to randomly select a sample and then calculate an estimate for the total basal area of all trees in the forest based on the sample. Each sampling method will be carried out  $10^5$  times, and then we compute the standard deviation of the  $10^5$  estimates for the totals. A lower standard deviation indicates a more accurate estimation method so is considered desirable.

You will be working from the data set `trees.csv` that has 3396 records, each with the following entries:

```
spp = species code
dbh = diameter at breast height, the trunk diameter 1.6 meters from the ground
ba = basal area of tree
volume = volume of wood
x = x-coordinate of tree location
y = y-coordinate of tree location
```

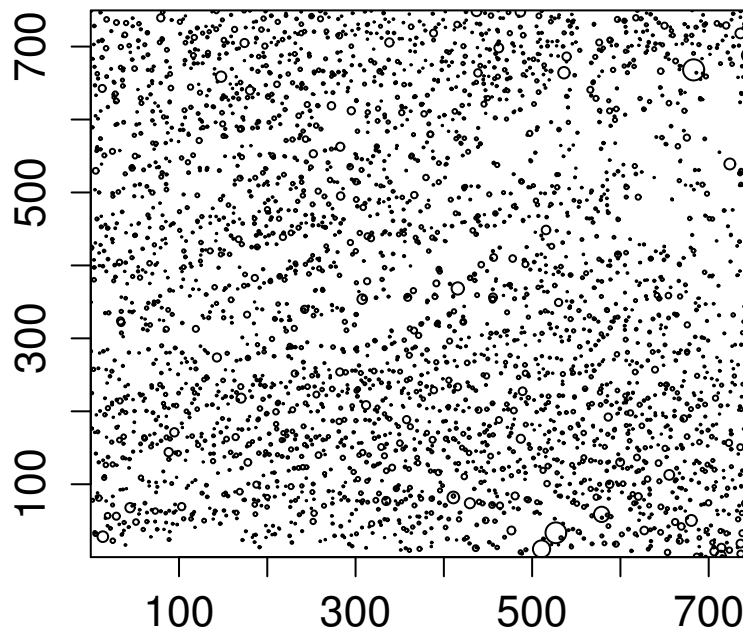


Figure 1: A 750-by-750 foot stand of 3396 trees. The area of each circle is proportional to the total basal area of the tree, which is the cross-sectional area of the trunk at about 2 meters from the ground.

## Part 1: Masuyama's Method<sup>1</sup>

For this method, we will select a random circular plot of radius  $r > 0$ . The center for the plot is a randomly selected point that is either in the stand or within a distance of  $r$  from the stand. For the purpose of this assignment, a random center  $(x, y)$  is selected uniformly from the region  $[-r, 750 + r] \times [-r, 750 + r]$ . (See Figure 2 for examples of plots.)

Once a random plot is selected, then all trees within the plot are selected for the sample, which we denote by  $S$ . Now let

$$a = \pi r^2, \quad A^* = (750 + 2r)^2$$

The probability that a tree  $z_i$  is selected into a random sample is given by

$$\pi_i = \frac{a}{A^*}$$

for every tree in the stand. We use the following formula to estimate the total basal area (TBA) for the entire stand:

$$\hat{t} = \sum_S \frac{y_i}{\pi_i} = \frac{A^*}{a} \sum_S y_i$$

where the sum runs over all trees in the sample  $S$ , and  $y_i$  is the basal area for tree  $z_i$ . For each tree,  $y_i$  is given in the data set by `ba`.

**Simulation** We will use simulation to determine the accuracy of each estimation method. First write a function to implement Masuyama's method,

```
[estBias estRmse ntrialVec] = p01Masuyama(ntrials, csvfile, r, plotsOn)
```

where `ntrials` is the number of random plots to generate, `csvfile` is the name of the file containing the tree data, `r` is the radius of the plots to generate, `stand` is a  $2 \times 2$  matrix where the first row corresponds to the minimum x and y coordinates of the stand, and the second row are the maximum x and y coordinates of the stand. The input `plotsOn` is a characters string to determine whether to generate plots (if it is 'on') or not. The function will output three vectors `estBias`, `estRmse`, and `ntrialVec` corresponding to the values of the estimate percentage bias and the estimate percentage rmse (see below) computed at intervals of the `ntrials` simulations, as defined in the corresponding elements of `ntrialVec`.

First set `r=37` (this gives a plot size of about 400 m<sup>2</sup>). Generate `ntrials = 105` random plots, and compute an estimate  $\hat{t}$  from each plot.

**Percentage Bias** This is given by

$$100 \frac{\text{average}(\hat{t}) - t}{t}$$

For this project,  $t = 311.906$  for the entire stand. The percentage bias gives a measure of how different the estimated values are (on average) from the true total, as a percentage of the total. In theory this is zero, but due to randomization in the simulations this value is typically between  $-0.5\%$  and  $0.5\%$ . (If it is not, then your code may have errors.)

**Percentage Root Mean Square Error** This is given by

$$100 \frac{\sqrt{\text{Var}(\hat{t})}}{t}$$

The root mean square error (RMSE) gives a measure of the variation in  $\hat{t}$  as a percentage of the total. It is desirable to have  $\hat{t}$  near  $t$ , so a smaller RMSE is better. We will use the RMSE to compare methods.

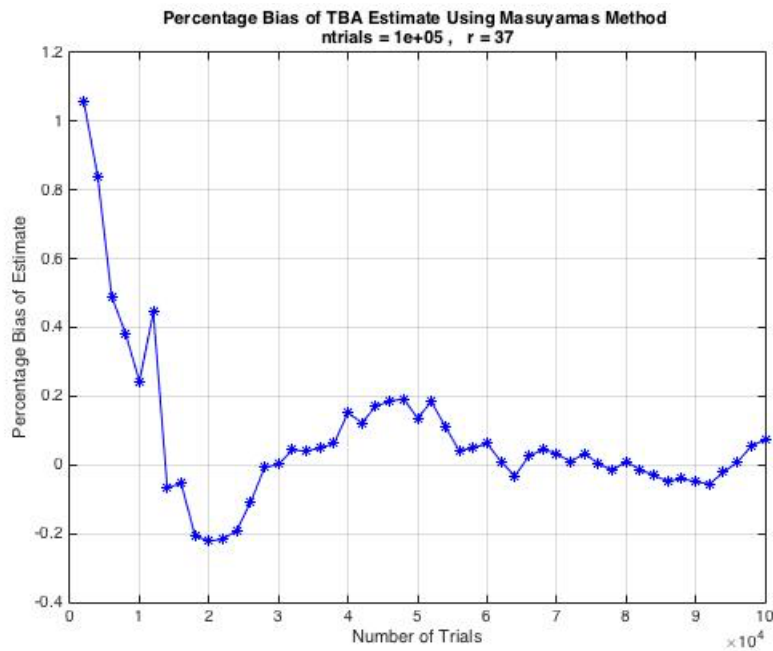
---

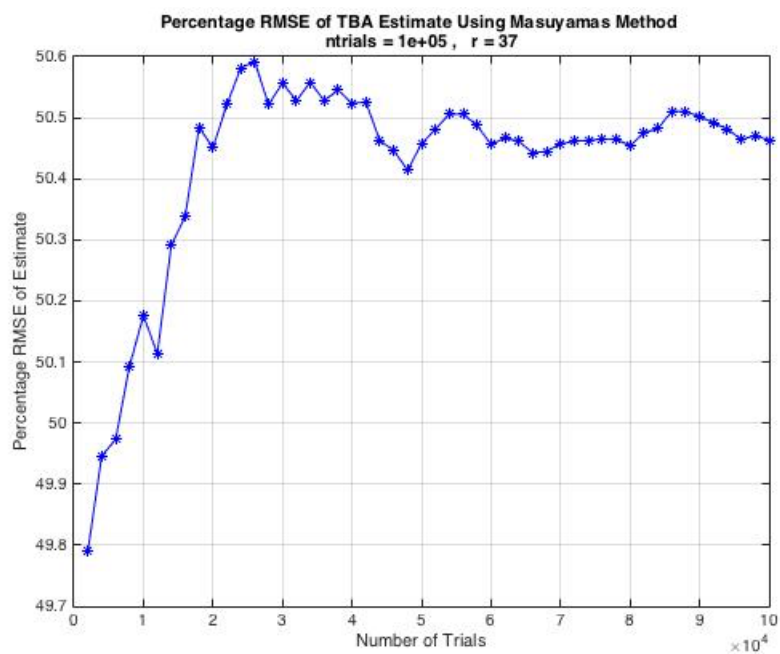
<sup>1</sup>Masuyama is the name of the statistician who proposed this method.

**Note:** In order to determine the trees whose centers are contained in a given plot, you should derive a boolean expression that will generate a logical vector to use to index the tree stat set. See the following link on logical indexing in Matlab:  
<https://www.youtube.com/watch?v=JE7I4Krj1PU>  
<http://blogs.mathworks.com/loren/2013/02/20/logical-indexing-multiple-conditions/>

The Mathematica users should use either `Select` or `MapThread` to accomplish the same thing as logical indexing in Matlab, as detailed in the following link:  
<http://mathematica.stackexchange.com/questions/2821/does-mathematica-have-advanced-indexing>

When generating `ntrials` plots, the `p01Masuyama` function will calculate the percentage bias and percentage rmse when the number of trials is a multiple of `nSample`. To calculate what `nSample` should be, you need to determine how large and integer `n` has to be such that  $\text{ntrials}/n < 50$ . That sampling interval will insure you compute and plot approximately 50 values of the percentage bias and percentage rmse, no matter how large `ntrials` might be. Example plots are shown below.

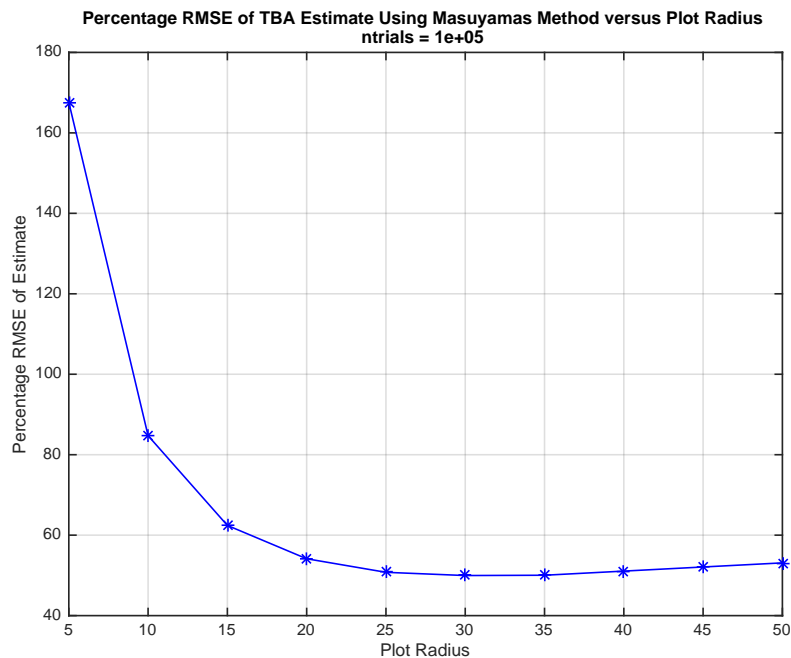




Once your `p01Masuyama` function has passed the tests, write a function,

```
[ ] = rmseVerR( ntrials,tbaMethod )
```

that take a TBA estimation function as the input `tbaMethod`, and computes the percentage RMSE for `ntrials` random plots for values of  $r = 5, 10, 15, \dots, 45, 50$  creates the the plot below.



For Part 1, you should submit your functions `p01Masuyama` and `rmseVerR`, as well as reproductions of the three plots above. Also speculate on the possible function that relates plot radius to RMSE.

## Part 2: Measure $\pi_i$ Method

This method is similar to Masuyama's, with two important differences:

1. Plot centers must be within the stand, not extending over the edge as with Masuyama.
2.  $\pi_i$  is not the same for all trees. Now we have

$$\pi_i = \frac{a(z_i)}{A}$$

where  $A$  equals the stand area ( $750^2$  for this project), and  $a(z_i)$  is the area of the portion of the plot of radius  $r$  centered at  $z_i$  (the location of tree  $i$ ) that overlaps the stand. In Figure

2, if the circles were centered on tree locations, then  $a(z_i)$  would be the area of the shaded regions. Trees at the edge of the stand have a smaller  $\pi_i$  than the rest.

The estimator for the Measure  $\pi_i$  method is the same as for Masuyama,

$$\hat{t} = \sum_S \frac{y_i}{\pi_i}$$

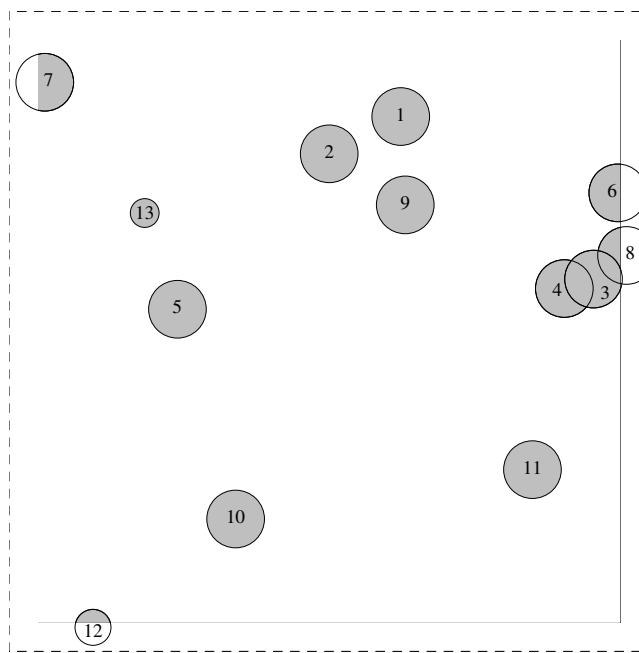


Figure 2: Example plots are numbered 1–11. (Ignore 12 and 13, which are smaller.) Note that some plots overlap the stand edge (solid line), and plot 8 is centered outside the stand. The dashed lines indicated the extended region of radius  $r$  where plot centers can be located.

To implement this estimator, write a function,

```
[estBias estRmse ntrialVec] = p02MeasurePi(ntrials, csvfile, r, plotsOn)
```

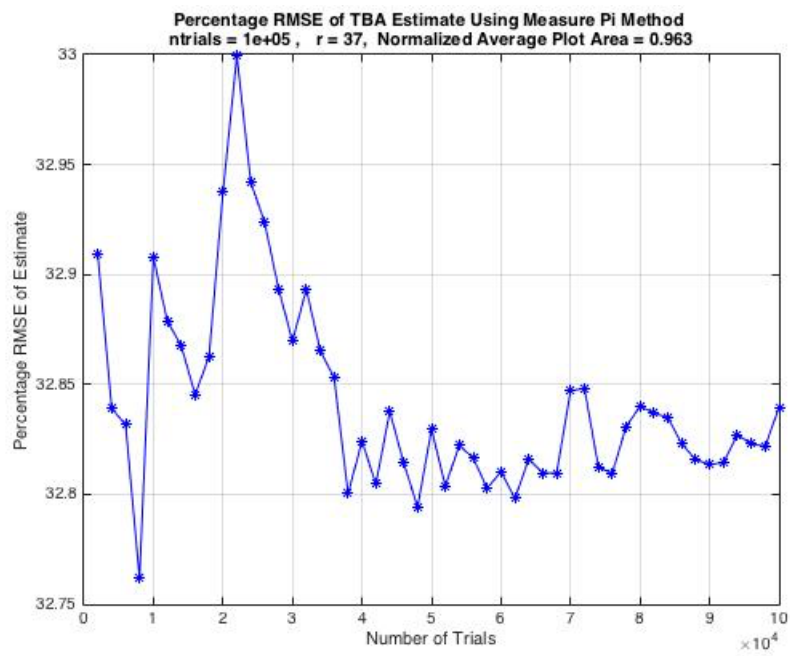
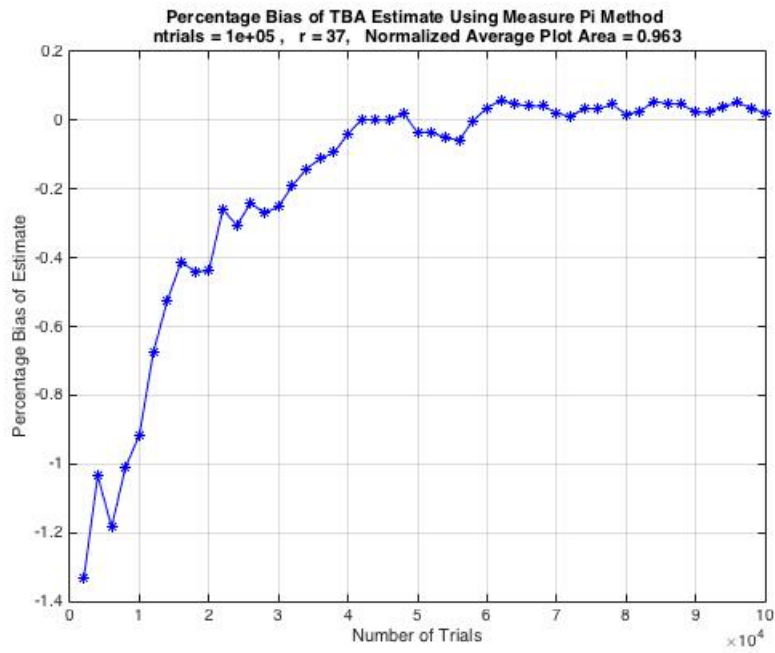
as in Part1 that uses the Measure  $\pi_i$  method to estimates the TBA. To determine the value of  $\pi_i$ , you will need to write a function

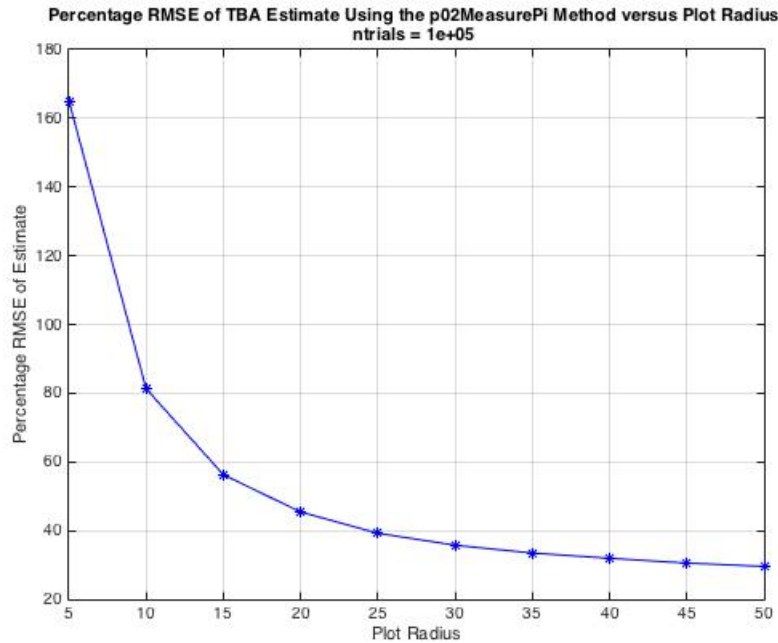
```
[area] = overlapArea(x,y,r)
```

that computes the intersection of a disk of radius  $r$  centered at coordinates  $(x,y)$  with the 750-by-750 stand of trees. This function will then be called from the function `p02MeasurePi`. See the `hints.pdf` file for directions on how to construct the `overlapArea` function.

Carry out a simulation similar to Part 1:  $r = 37$  with  $10^5$  simulated samples.

1. Compute the percentage bias and percentage rmse for this estimator, and generate and save plots as shown below.
2. Determine the average area of a plot (normalized by the area of a plot of radius  $r$ ) in your simulation and include it as part of the titles in your plots as show below.
3. Use the function `rmseVerR` from Part 1 to plot the RMSE versus the radius for the Measure  $\pi_i$  method.





For Part 2, you should submit your functions `p02MeasurePi`, as well as reproductions of the three plots above.

### Part 3: Repeated Masuyama

This method is similar to Masuyama in Part 1, but if a sample plot overlaps outside the stand boundary, then we generate another smaller sample plot. The second plot has radius selected so that its area is equal to the amount of overlap outside the stand of the first plot. This approach is repeated until a plot is obtained that does not overlap the boundary. The selection method is such that the combined stand area within the plots is equal to  $a$ . All trees in the combined plots are included in the sample, with trees included in several plots counted once for each plot they are in. Figure 3 shows an example of plot selection.

The estimator for repeated Masuyama is

$$\hat{t} = \frac{A}{a} \sum_{S^*} y_i$$

where  $S^*$  indicates all trees in the sample plots, with trees in multiple plots counted once per plot.

Write a function,

```
[estBias estRmse ntrialVec] = p03RepeatedM(ntrials, csvfile, r, plotsOn)
```

as in Part1 that uses the repeated Masuyama method to estimates the TBA. To implement this method, you will again need the function

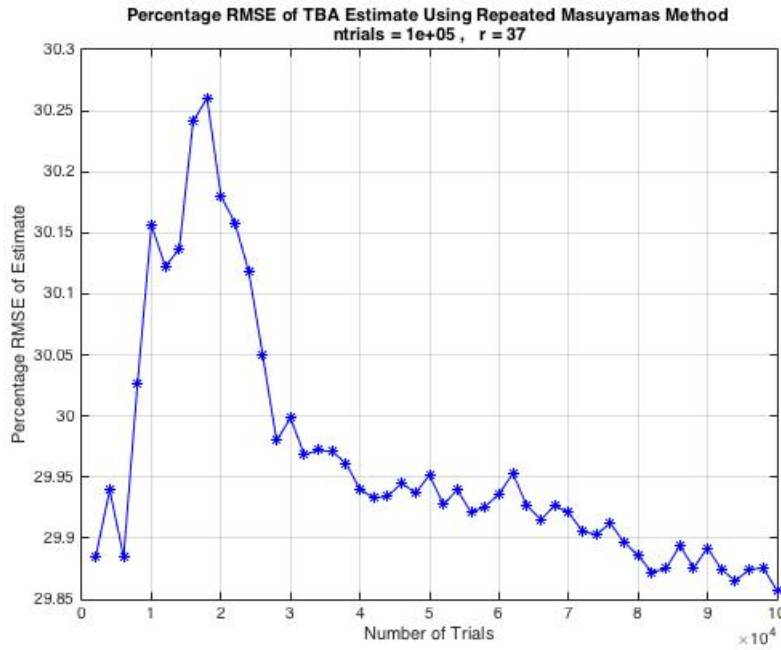
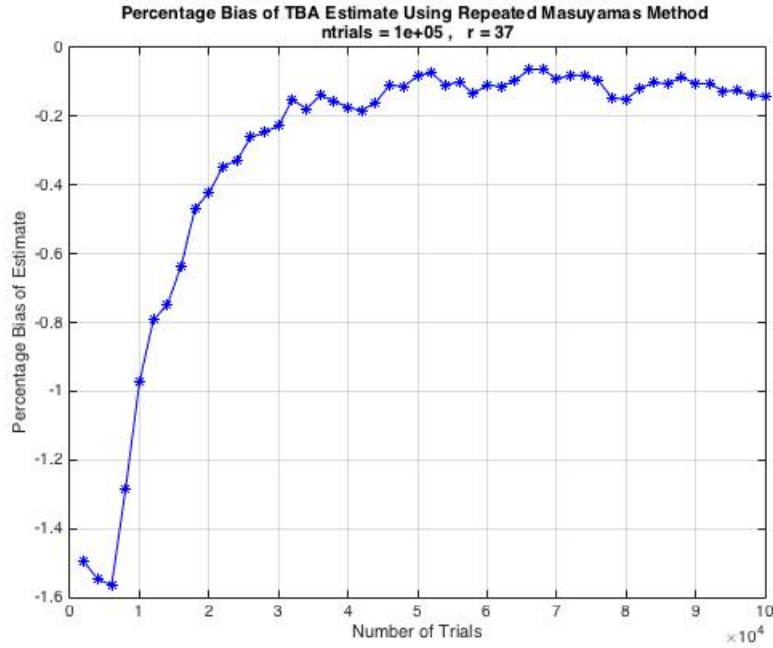
```
[area] = overlapArea(x,y,r)
```

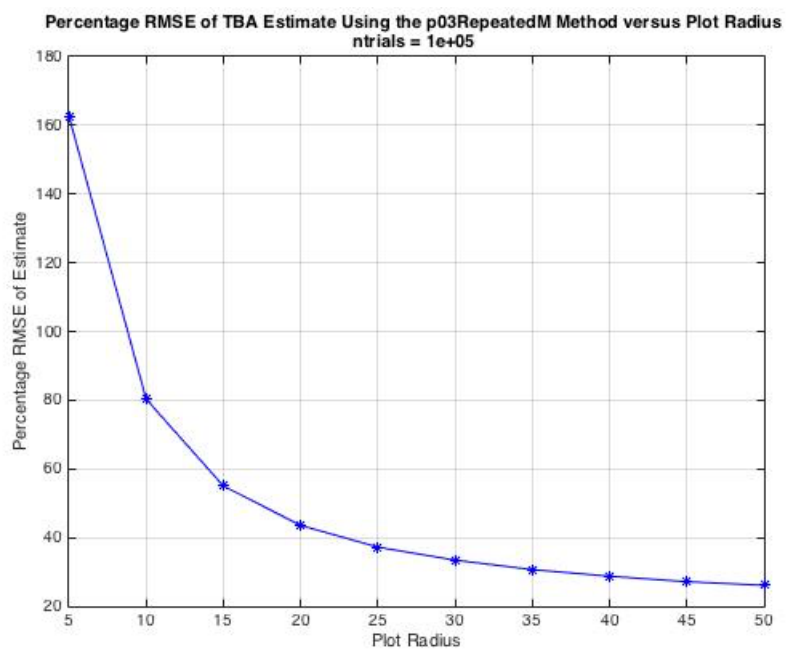
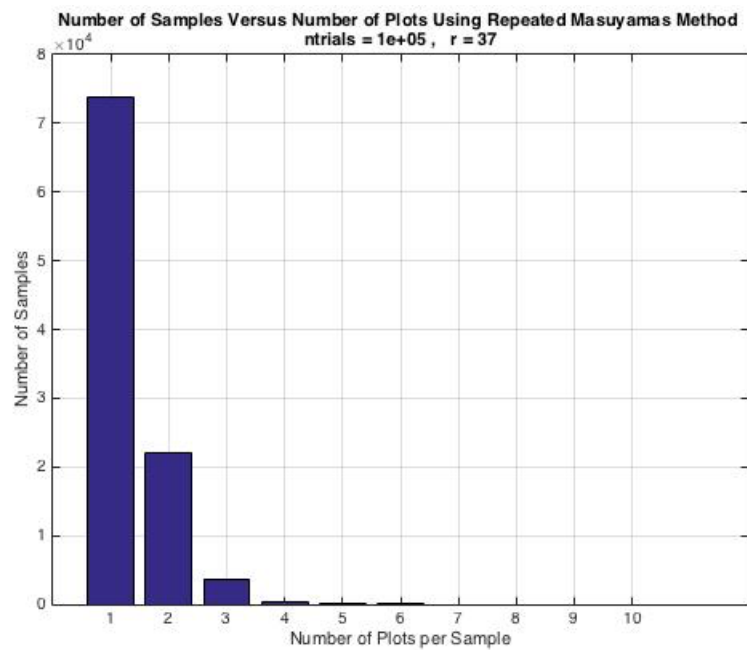
that computes the intersection of a disk of radius `r` centered at coordinates `(x,y)` with the 750-by-750 stand of trees. In this case the tree centers can be outside the boundary of the stand as in Part 1. See the `hints.pdf` file for directions on how to construct the `overlapArea` function.



As with the previous parts, use  $r = 37$  and  $10^5$  simulations.

1. Compute the percentage bias and percentage rmse for this estimator, and generate and save plots as shown below.
2. Tabulate the number of samples with one plot, two plots, three plots, and so on and display it as a bar plot as shown below.
3. Use the function `rmseVerR` from Part 1 to plot the RMSE versus the radius for the Measure pi method.





For Part 3, you should submit your functions `p03RepeatedM`, as well as reproductions of the four plots above.

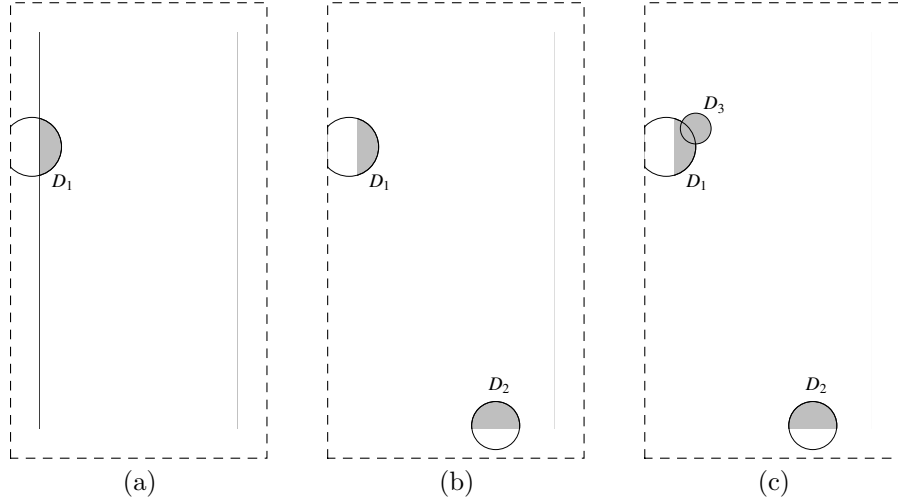


Figure 3: An example of an repeated Masuyama sample. For parts (a)–(c), the stand  $\mathcal{A}$  is the region enclosed by the solid rectangle and  $\mathcal{A}^*$  is the region enclosed by the dashed rectangle. (a) The first sample plot is  $D_1$ , which has total area equal to  $a = \pi r^2$ . (b) The second sample plot is  $D_2$ , which has total area equal to the portion of  $D_1$  outside  $\mathcal{A}$ . (c) The third sample plot is  $D_3$ , which has total area equal to the portion of  $D_2$  outside of  $\mathcal{A}$ . Since  $D_3$  is entirely within  $\mathcal{A}$ , it is the last plot in the sample. The sample consists of the population elements contained in the shaded regions. The total area of the portions of the plots inside  $\mathcal{A}$  is  $a$ . For this sample, trees in the intersection of  $D_1$  and  $D_3$  will be counted twice in the estimate.