R Graduate Project

Fisher Ankney

11/15/2018

I've created a function to help solve a very specific probability problem, estimating the solution through one million simulations. Recently, in probability theory we were posed with the following problem.

Two friends arrive at a specific spot between 10:00 AM and 11:00 AM. Suppose that each friend arrives randomly during this hour, and each waits 10 minutes. What is the probability that they meet?

This problem can be solved traditionally using the joint probability distribution, and integrating over the proper bounds. Given that the arrival time of each friend is a continuous uniform distribution, the answer can be derived from the following equation.

$$2 * \left(\int_0^{50} \int_x^{x+10} \frac{1}{3600} \, dy \, dx + \int_{50}^{60} \int_x^{60} \frac{1}{3600} \, dy \, dx \right) = \frac{11}{36}$$

My R program estimates this probability through one million vectorized simulations, done in a fraction of a second. Not only can it estimate this probability, it can estimate the probability of two friends meeting within any interval of time, and waiting for any period of time.

The inputs to my function are interval, and wait_time. Interval defaults to the above problem, 60 (minutes). wait_time also defaults to the above problem, 10 (minutes). Select any non-negative integer for these arguments.

The outputs of the function are a probability estimate of the two friends meeting, and a graphic representation of this estimate. The graphic displays a histogram of the difference in arrival time for the two friends. So, if one friend arrives at 10:30, and the other at 10:35, a time of +5 is plotted on the histogram. This is repeated for all one million simulations. Two red lines indicating the maximum wait time (positive and negative) are plotted to illustrate the area in which the friends will meet; the probability that is being estimated. These vertical red lines are responsive to the wait_time argument the user inputs.

The following are examples of my function under different circumstances.

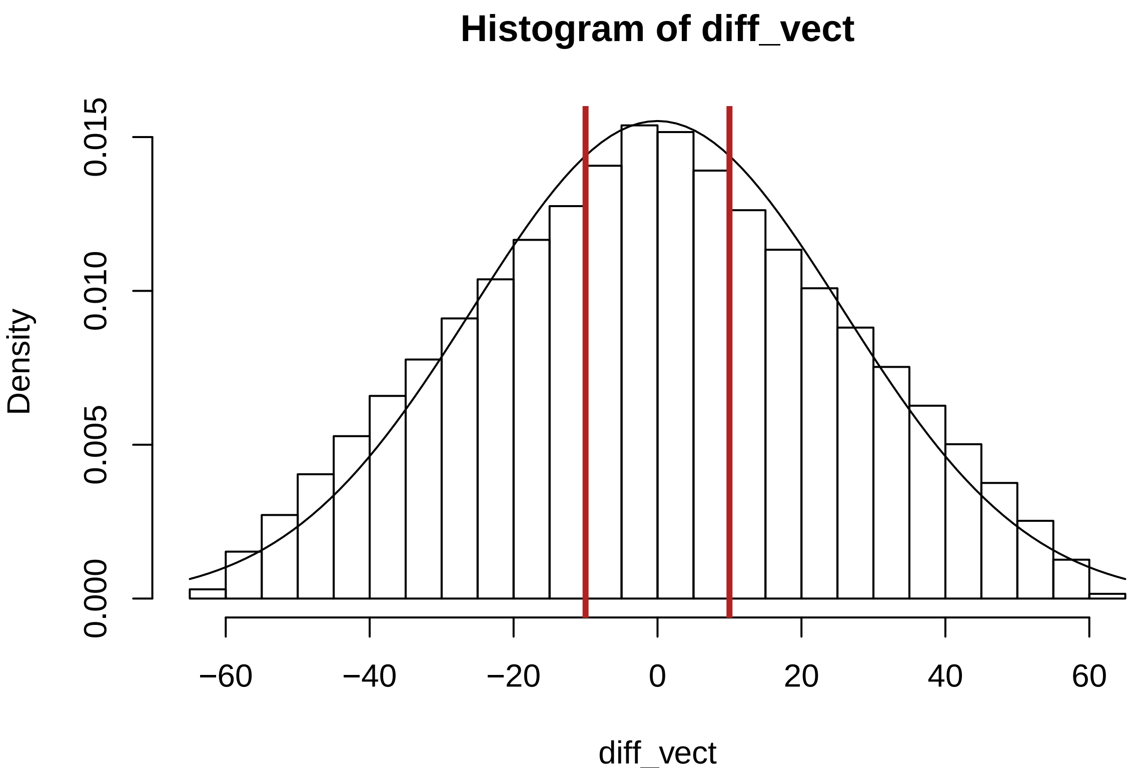
Executed Code Demonstration

First I'll source the function.

```
setwd('~/Documents/data_science/r_stat_5193/scripts')
source('AnkneyProject.R', print.eval = T)
```

Next, I'll test the function on the previously described case.

```
friends_meet(60, 10)
```

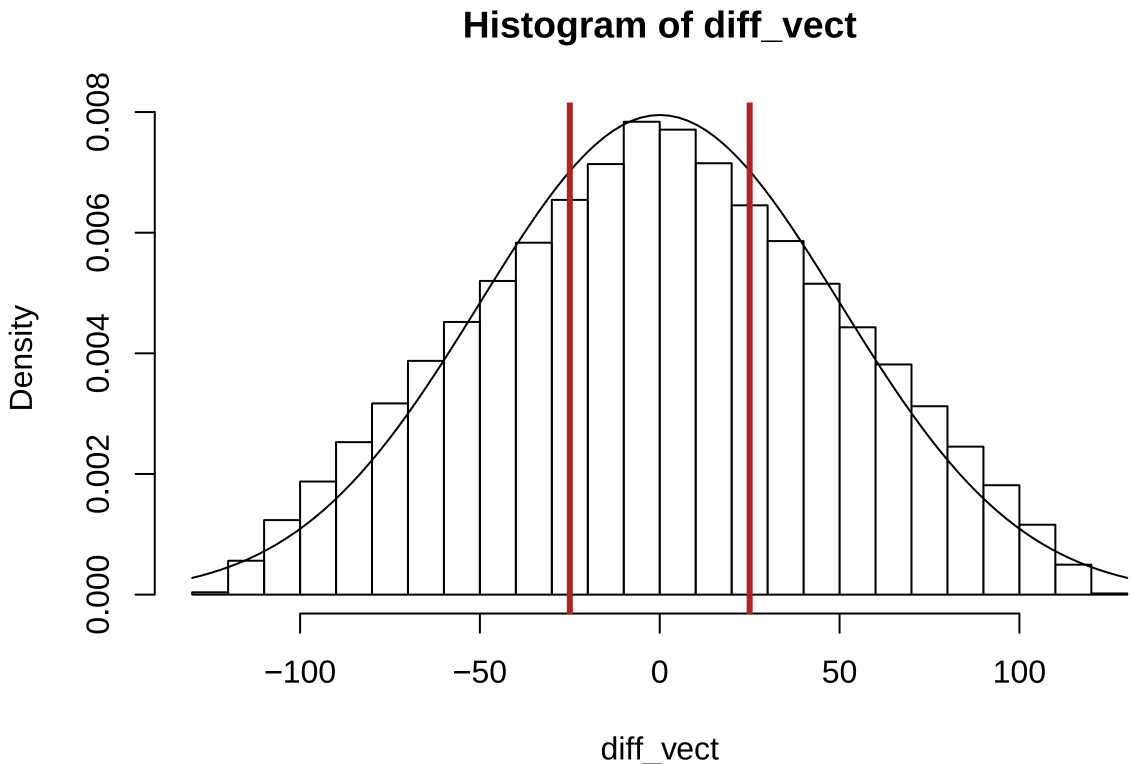



```
## [[1]]
## NULL
##
## [[2]]
## [1] 0.305901
```

Here you can see that my function estimates the actual probability to within 0.0005 of $\frac{11}{36}$. The associated graphical output plots the difference between the two friends meeting, which ends up being a normal distribution centered at zero. The possibility of the two friends meeting occurs between the two red lines.

Let's use the function again on several different scenarios to prove that it's responsive.

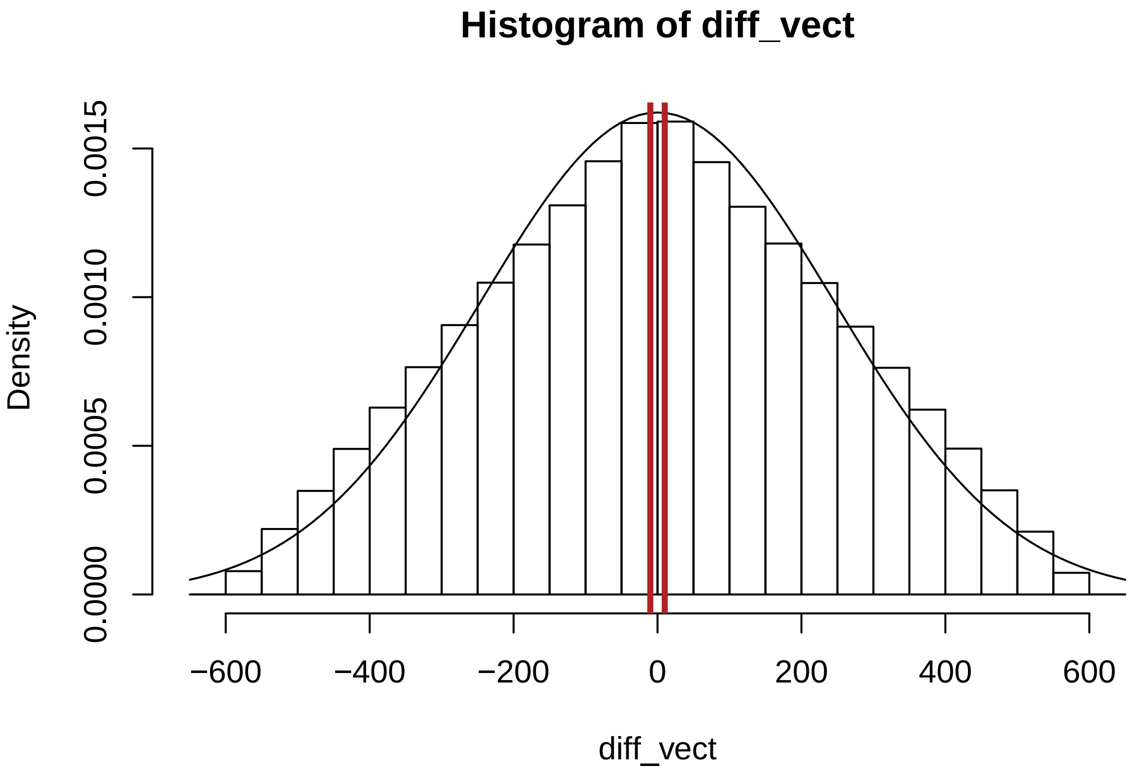
```r
# friends meet in a 120-minute interval, and wait 25 minutes each
friends_meet(120, 25)
```




```
## [[1]]
## NULL
##
## [[2]]
## [1] 0.37109
```

A probability of 0.37 arises from this scenario.

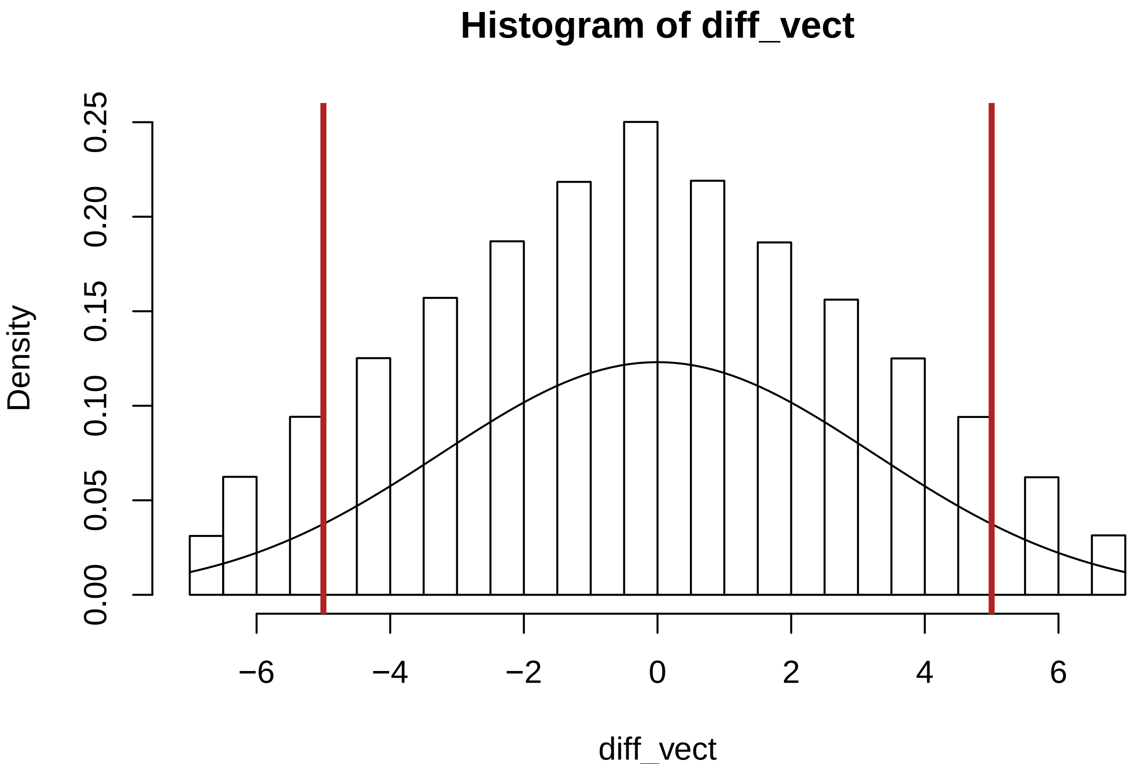
```r
# friends meet in a 10-hour (600 minute) interval, and wait 10 minutes each
friends_meet(600, 10)
```




```
## [[1]]
## NULL
##
## [[2]]
## [1] 0.034744
```

A 0.03 chance of the two friends meeting under these circumstances.

```r
# edge case - friends decide to meet in a 5 minute interval,
# each waits 5 minutes
friends_meet(5,5)
```

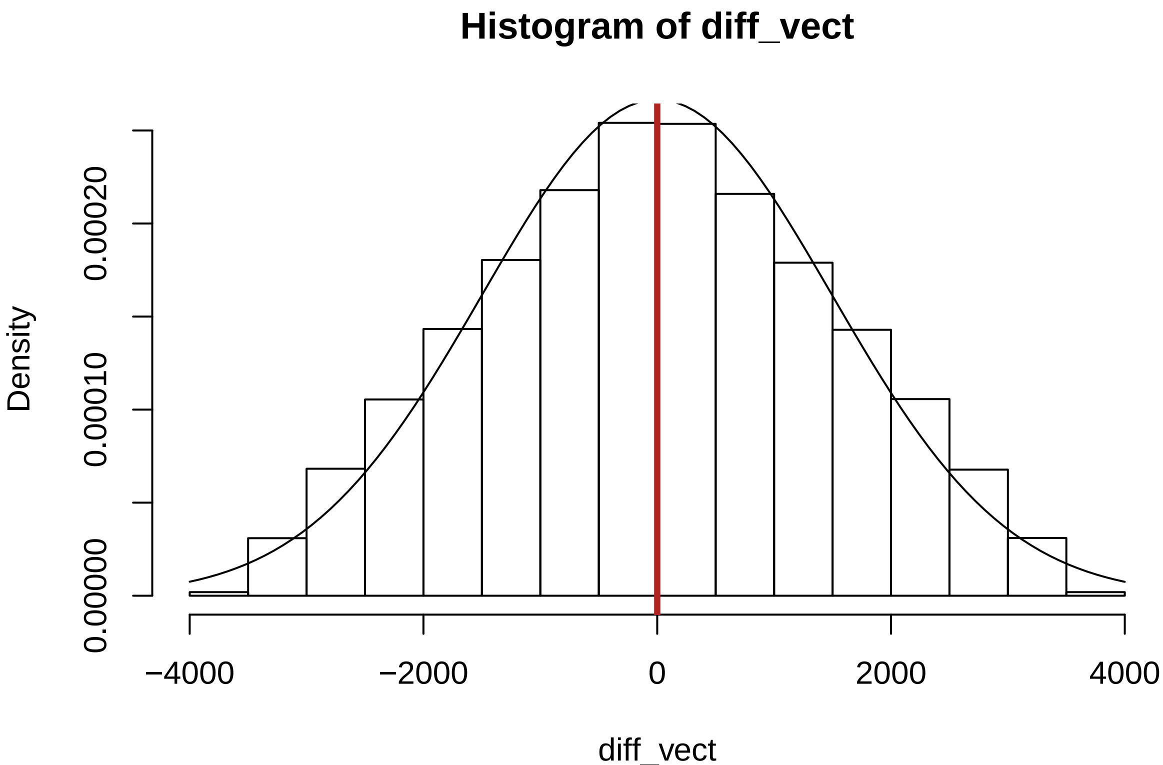



```
## [[1]]
## NULL
##
## [[2]]
## [1] 0.906401
```

You may think that this simulation is wrong, if there are only 5 minutes to meet, and each friend waits for 5 minutes, mustn't they meet? It is likely they meet, but not guaranteed. If friend A arrives at 10:00 AM exactly, and waits for 5 minutes exactly, they will leave at 10:05 AM, and no seconds. The nature of my simulation, and this problem, allows for Friend B to arrive at 10:05.59, thus not meeting with the first friend. This adjustment is necessary to report the proper probability in the initial problem, but can easily be adjusted if this isn't the case in subsequent problems.

```r
# edge case - friends decide to meet in a one day interval (3660 minutes)
# both friends wait 0 minutes after their arrival minute.

friends_meet(3660,0)
```




```
## [[1]]
## NULL
##
## [[2]]
## [1] 0.000282
```

In this final case, the only way for the two friends to meet in the entire day, is for them to arrive at the exact same moment. A highly unlikely, but not impossible, case.

This concludes my explanation and demonstration of the `friends_meet()` function in R. Please see the attached R script to read into the function, and perhaps test it out for yourself.