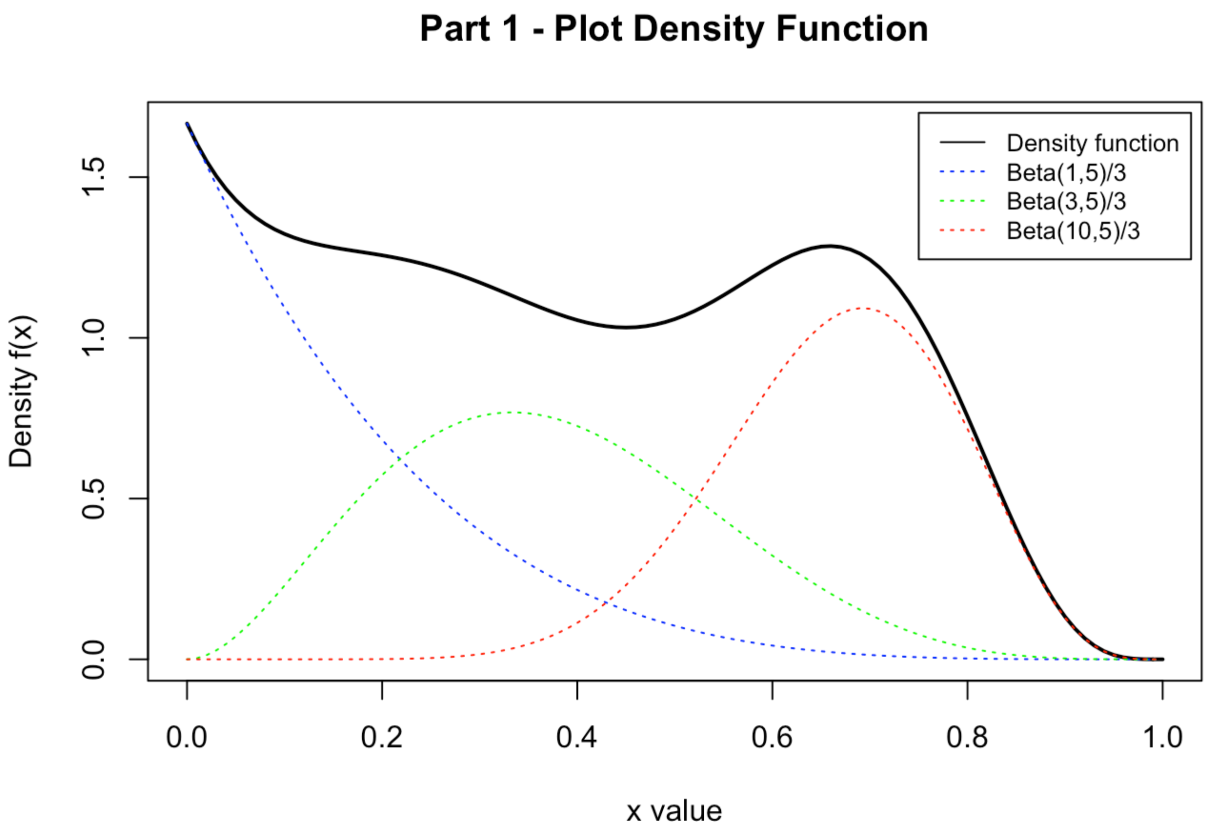
Assessment 2: Monte Carlo Algorithms

1 AUGUST, 2021
PAUL JOHN CRONIN
Z5330951

1. Plot the density f(x)

The following plot describes the frequency distribution of methylation. Please note that the density function is comprised of three Beta functions, where one of three does not approach zero smoothly around x=0.



2. Implement an accept/reject algorithm

We are asked to determine the value K, and its associated x value.

The value of K = 1.6667
The position of K is at x = 0

3. Compute the observed and theoretical acceptance rate

We are asked to compute the acceptance rate and theoretical acceptance rate.

The acceptance rate is: 0.5978
Theoretical acceptance rate is: 0.6

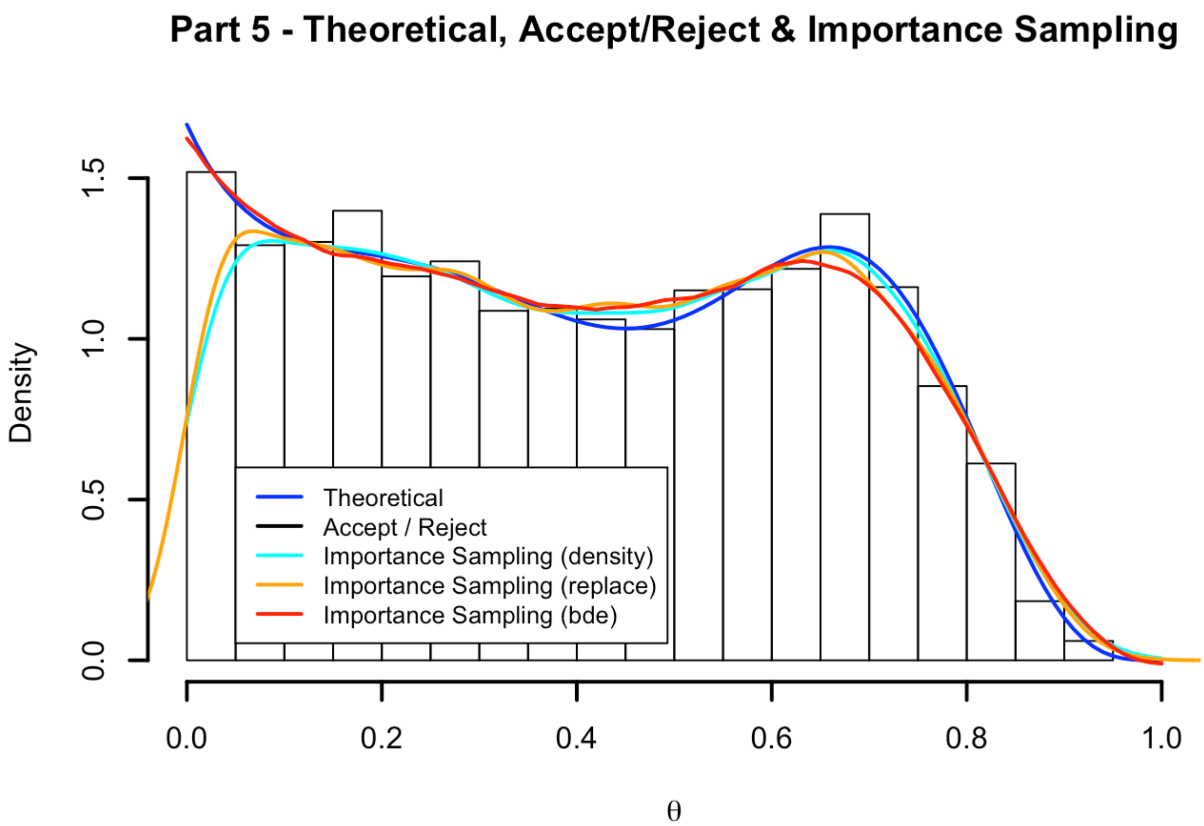
4. Implement an importance sampling algorithm

This same values as used in the accept / reject algorithm are used in the R Markdown code to derive the importance sampling weights.

5. Compare theoretical, A/R and IS on same plot

On the plot below are the theoretical density (blue), the density of the accepted values for the accept/reject algorithm (black histogram), and the density of the values weighted by three importance sampling algorithms (cyan, orange & red).

It is important to note the divergence from the theoretical density for two versions of the importance sampling algorithms (density & replace), but not accept / reject.



6. Which of the two algorithms seems to approximate the target distribution better?

In class notes, webinars, and online references, we are taught that there is no perfect algorithm. To evaluate which is the better algorithm in this case, Monte

Carlo error convergence plots were computed as well as the quantile data was evaluated. Another important point to evaluate is the anomaly around x=0.

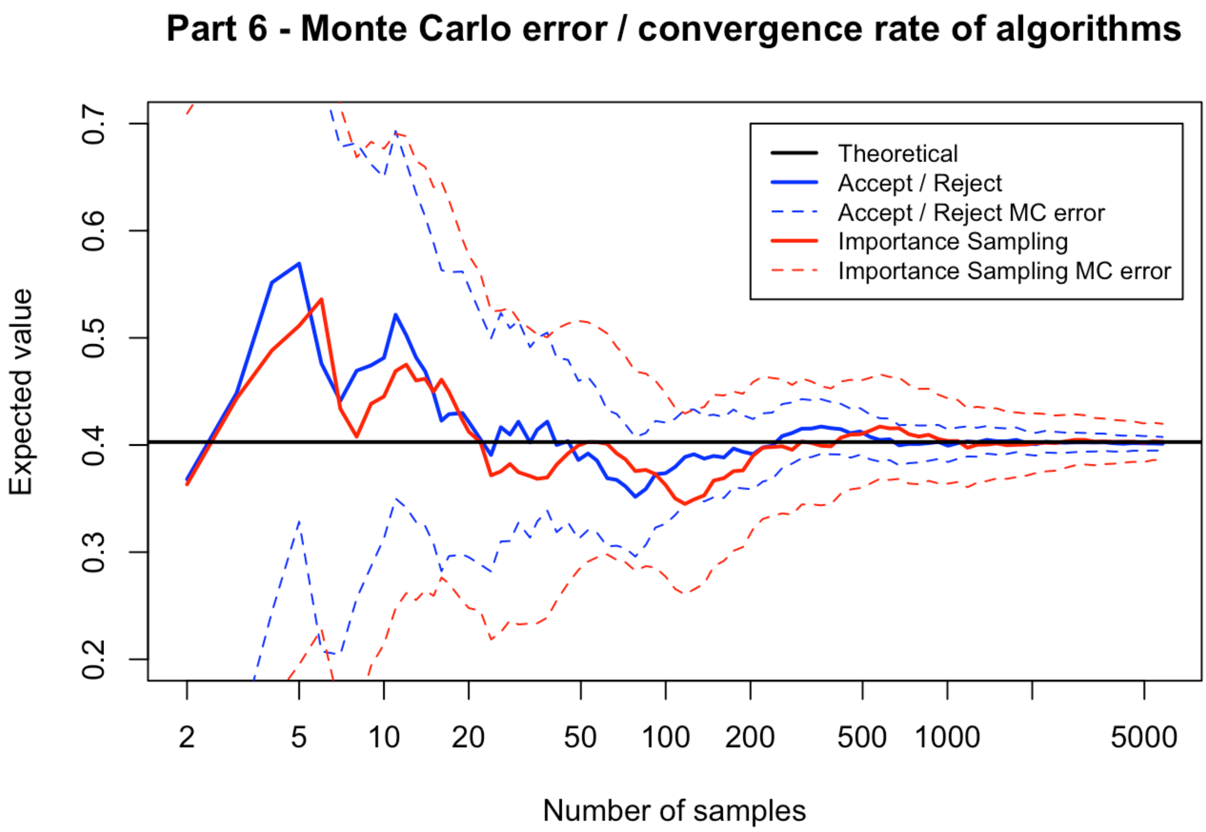
Anomaly at x=0 (from Part 5)

Examining the density function around x=0, we find that it does not go smoothly to zero. This effectively acts as a discontinuity, causing the importance sampling algorithms (density & replace) to have a significant failure at that point.

This anomaly can be remedied with the use of the BDE algorithm, however, I would argue that even this is somewhat inappropriate, as I had to carefully choose the value for “b” to obtain a smooth curve, approximating the theoretical

Solution convergence rate

As expected, both the accept/reject algorithm and the importance sampling algorithm converge to the theoretical expected value of 0.4028, however the accept / reject algorithm has a smaller Monte Carlo error and converges faster than the importance sampling algorithm.



Quantile Data

The following table shows the quantile data for the theoretical distribution, the accept/reject algorithm and the importance sampling algorithm. For nearly all quantiles, the accept/reject algorithm has a smaller error to the theoretical value than the importance sampling algorithm.

Please note that the theoretical values are derived from Mathematica – please see the Appendix.

Quantile	Theoretical value	Accept / Reject	Importance Sampling
0 %	0.0000	0.0003418 (0.34% error)	0.0003418 (0.34% error)
25%	0.1814	0.1827 (0.71% error)	0.1846 (1.78% error)
50 %	0.3937	0.3935 (0.05% error)	0.4001 (2.82% error)
75 %	0.6207	0.6194 (0.22% error)	0.6185 (0.36% error)
100 %	1.0000	0.9443 (5.57% error)	0.9536 (4.64% error)

Conclusion

After evaluating three different factors, those being:

- the anomaly caused by the discontinuity at x=0 with the importance sampling algorithm,
- the fact that the accept / reject algorithm converges faster to the true solution than the importance sampling algorithm, and
- the quantile values have less error for the accept/reject than for the importance sampling algorithm,

then I conclude that, in this case, the accept/reject algorithm is superior to the importance sampling algorithm ***in this particular case***.

Appendix

Mathematical Code for theoretical quantiles

```
In[95]:=
TheorQuantiles =  Integrate[(PDF[BetaDistribution[1, 5], x] + PDF[BetaDistribution[3, 5], x] +
PDF[BetaDistribution[10, 5], x])/3, {x, 0, a}, Assumptions -> 0 <= a <= 1, Assumptions -> a \[Element] Reals];

In[96]:=
Solve[TheorQuantiles[[1, 1, 1]] == 0.0, a, Assumptions -> 0 <= a <= 1, Assumptions -> a \[Element] Reals]
Solve[TheorQuantiles[[1, 1, 1]] == 0.25, a, Assumptions -> 0 <= a <= 1, Assumptions -> a \[Element] Reals]
Solve[TheorQuantiles[[1, 1, 1]] == 0.5, a, Assumptions -> 0 <= a <= 1, Assumptions -> a \[Element] Reals]
Solve[TheorQuantiles[[1, 1, 1]] == 0.75, a, Assumptions -> 0 <= a <= 1, Assumptions -> a \[Element] Reals]
Solve[TheorQuantiles[[1, 1, 1]] == 1, a, Assumptions -> 0 <= a <= 1,  Assumptions -> a \[Element] Reals]

Out[96]= {{a -> 0.}, {a -> 0.}}
Out[97]= {{a -> 0.181397}}
Out[98]= {{a -> 0.393747}}
Out[99]= {{a -> 0.620717}}
Out[100]= {{a -> 1}, {a -> 1}, {a -> 1}, {a -> 1}, {a -> 1}}
```