ECS256 - Homework III

Olga Prilepova, Christopher Patton, Alexander Rumbaugh,
John Chen, Thomas Provan

February 17, 2014

Problem 1.a

First, we'll derive π_i. The definition of the tree searching markov model leads to the following set of balance equations for the long-run state probabilities:

$$\pi_i = \pi_{i-1} q_{i-1} = \pi_0 \prod_{j=0}^{i-1} q_j \quad \text{for } i \geq 1, \text{ and}$$

$$\pi_0 = \sum_{i=1}^{\infty} \pi_i (1 - q_i) \quad \text{for } i = 0.$$

This definition for π_0 is a bit unweildy. Since the chain is positive recurrent, we can also think of this quantity as one over the expected recurrance time, as in eq. (10.63) in the book:

$$\pi_0 = \frac{1}{E(T_{0,0})}$$

$$E(T_{0,0}) = 1 + \sum_{k \neq 0} p_{0,k} E(T_{k,0}) \quad \text{By eq. (10.65)}$$

$$= 1 + p_{0,1} E(T_{1,0})$$

$$= 1 + p_{0,1}(1 + \sum_{k \neq 0} p_{1,k} E(T_{k,0}))$$

$$= 1 + p_{0,1}(1 + p_{1,2} E(T_{2,0}))$$

$$= 1 + p_{0,1}(1 + p_{1,2}(1 + \sum_{k \neq 0} p_{2,k} E(T_{k,0})))$$

$$= 1 + p_{0,1}(1 + p_{1,2}(1 + p_{2,3} E(T_{3,0})))$$

and so on. This unravels into a familiar closed form:

$$E(T_{0,0}) = 1 + q_0(1 + q_1(1 + q_2(1 + \ldots) \ldots)))$$

$$= 1 + q_0 + q_0 q_1 + q_0 q_1 q_2 + \ldots$$

$$= 1 + \sum_{i=1}^{\infty} \left[\prod_{j=0}^{i-1} q_j \right]$$

If the model is positive recurrent, then there exists some value R such that

$$R = \sum_{i=1}^{\infty} \left[\prod_{j=0}^{i-1} q_j \right] < \infty.$$

Thus,

$$\pi_i = \frac{\prod_{j=0}^{i-1} q_j}{1 + R} \quad \text{for } i \geq 0.$$

Next, $E(T_{i,0})$ follows a similar pattern.

$$
\begin{aligned}
E(T_{i,0}) &= 1 + \sum_{k \neq 0} p_{i,k} E(T_{k,0}) \\
&= 1 + p_{i,i+1} E(T_{j+1,0}) \\
&= 1 + q_i + q_i q_{i+1} + q_i q_{i+1} q_{i+2} + \ldots \\
&= 1 + \sum_{j=i}^{\infty} \left[\prod_{k=i}^{j} q_k \right].
\end{aligned}
$$

Problem 1.b

If $q_i = 0.5$ for all i, then R is a geometric series that indeed converges.

$$\pi_2 = \frac{0.5 \cdot 0.5}{1 + \sum_{i=1}^{\infty} 0.5^{i-1}} = \frac{0.25}{1 + 2} \approx \boxed{0.083.}$$

$$E(T_{2,0}) = 1 + \sum_{j=2}^{\infty} 0.5^{j-2} = 1 + \sum_{j=1}^{\infty} 0.5^{j-1} = 1 + 2 = \boxed{3.}$$

Problem 1.c

The rate of backtracking, in terms of the stationary probabilities π_i, is simply

$$\sum_{i=1}^{\infty} \pi_i (1 - q_i).$$

Problem 2.a[1]

In this problem, we are given the task of generating a method-of-stages approximation of a distribution, given its quantile function. To accomplish the approximation, we seek to combine a set of erlang distributions and receive the approximation as the sum of the erlang distributions.

Problem 2.b

Using the **ermixobj** object generated in Problem 2.a, we are able to generate a set of **nmix** erlang distributions with parameters given as:

$$\text{Shape} = \mathbf{r}$$
$$\text{Rate} = \mathbf{lamb}$$

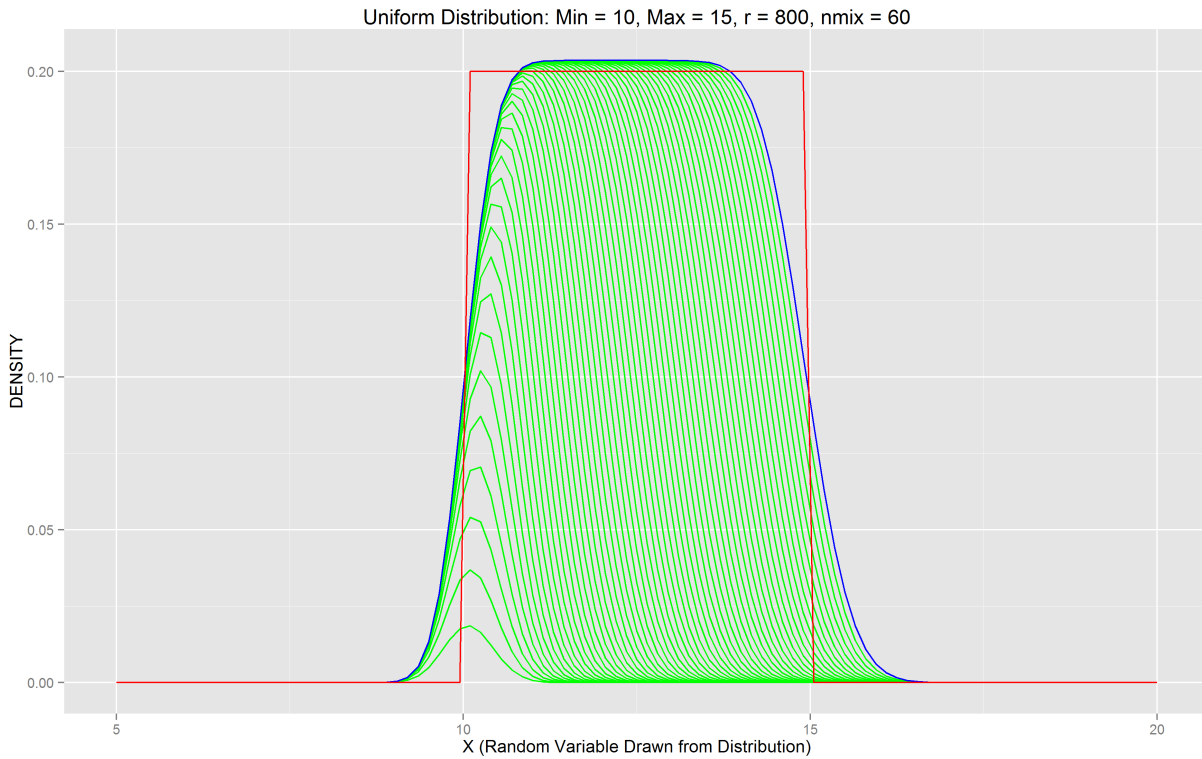The combination of all nmix erlang distributions yields our method-of-stages approximation of the quantile function fed into erlangmix() in Problem 2.a.

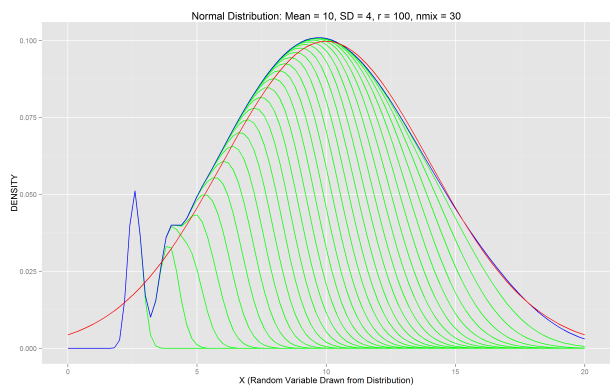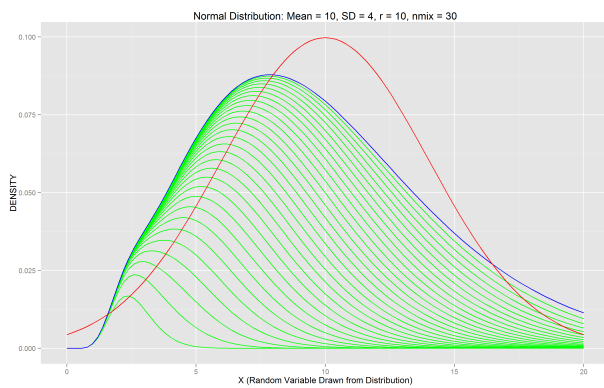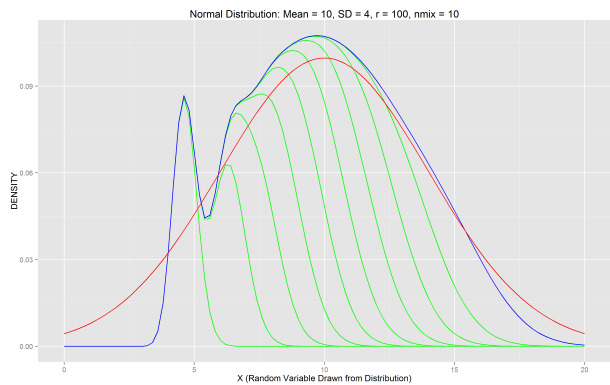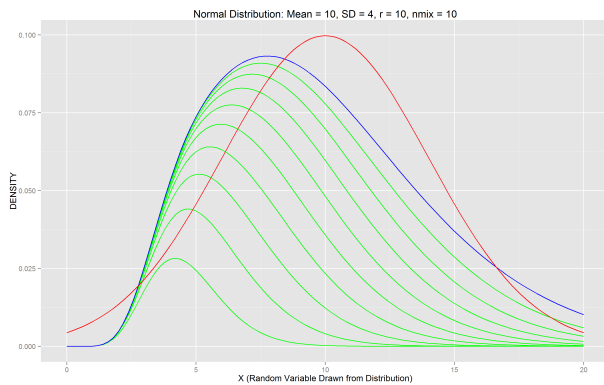Here, we explored the effect of different values of r and nmix on the approximation.
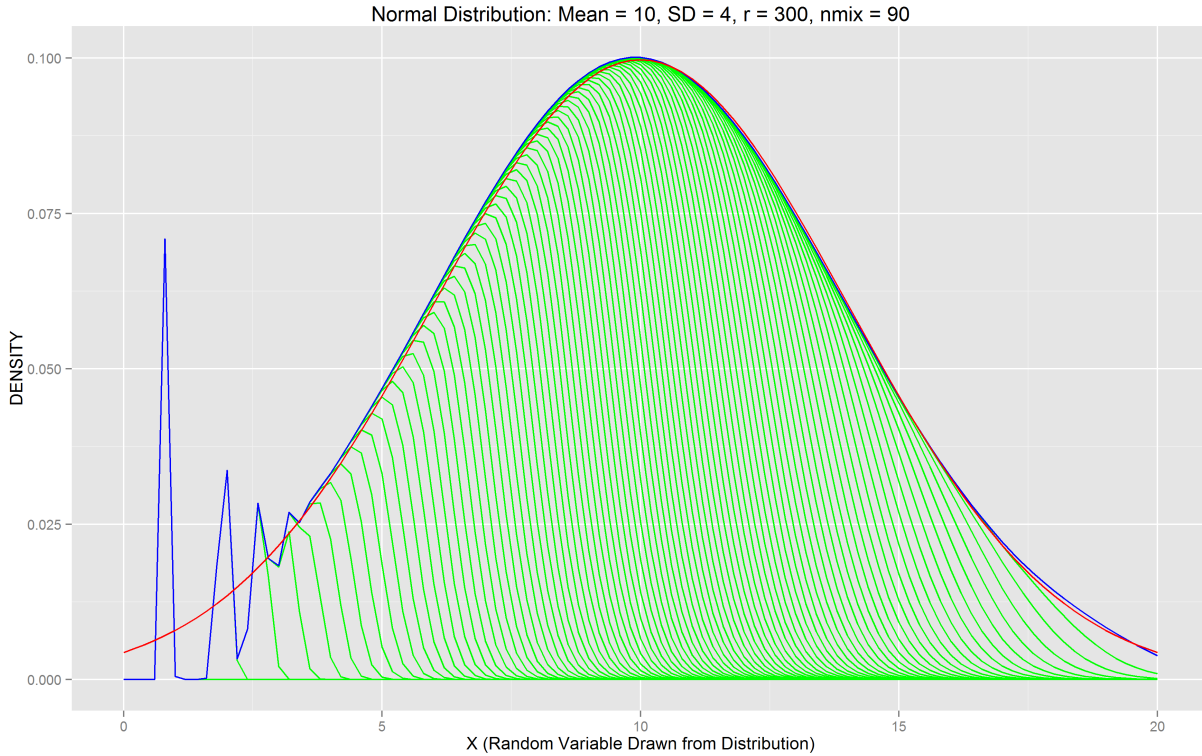
For a uniform distribution with minimum = 10, maximum = 15:

[1]See `ErlangMix.R`.

Uniform Distribution: Min = 10, Max = 15, r = 800, nmix = 60
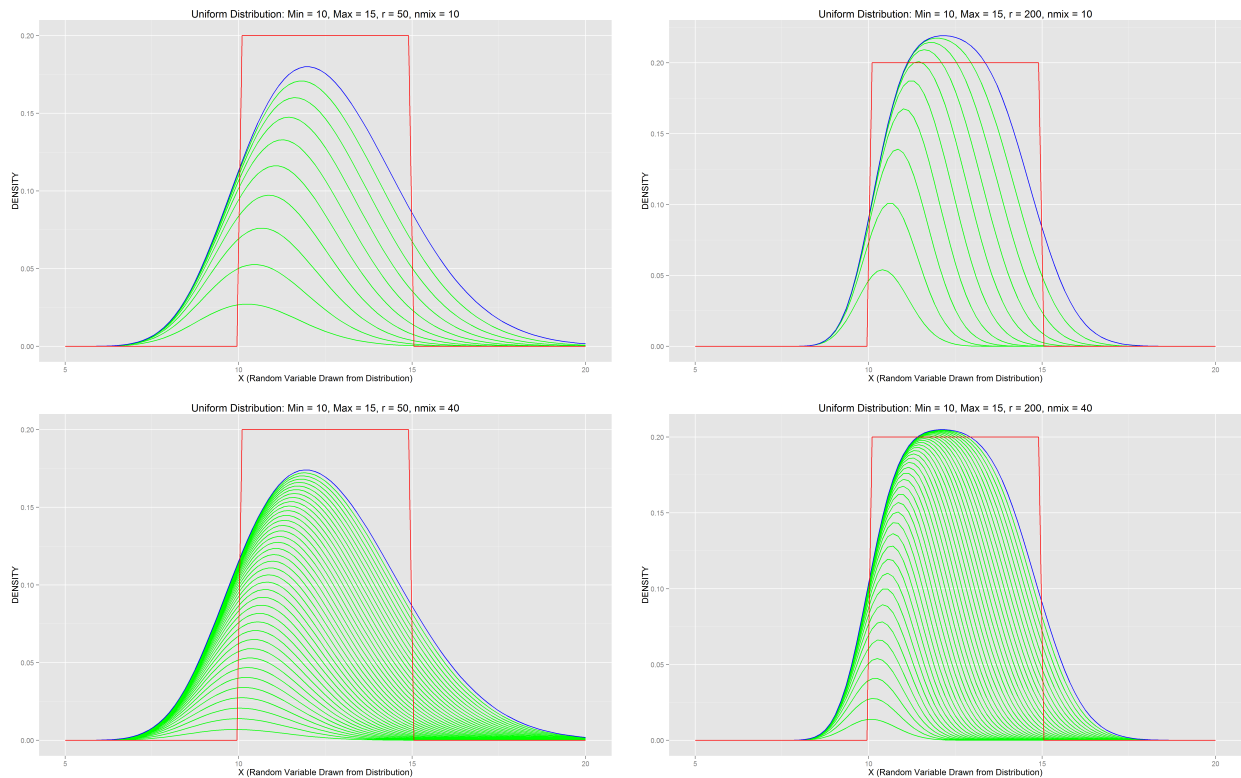
For a normal distribution with mean = 10, standard deviation = 4:

Normal Distribution: Mean = 10, SD = 4, r = 10, nmix = 10

Normal Distribution: Mean = 10, SD = 4, r = 100, nmix = 10

Normal Distribution: Mean = 10, SD = 4, r = 10, nmix = 30

Normal Distribution: Mean = 10, SD = 4, r = 100, nmix = 30

Normal Distribution: Mean = 10, SD = 4, r = 300, nmix = 90

The plots show that increasing textbfr reduces the magnitude of the error of our approximation, while textbfnmix controls the resolution of the approximation. By increasing both, we can get an increasingly accurate approximation of the given distribution.

Problem 3.a

See `HtoF.R`.

```
1  htof = function(hftn,t,lower){
2    density_val = c()
3    for(val in t)
4    {
5      density_val = c(density_val, hftn(val) *
6            exp(-1*integrate(hftn,lower,val)$value))
7    }
8    return (density_val)
9  }
```

Problem 3.b

Given a hazard function, $h(t)$, the density function, $f(t)$, can be found as follows:
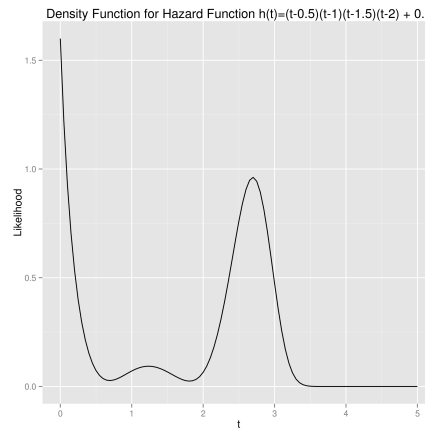
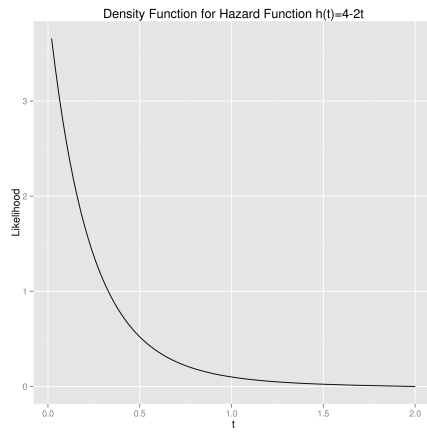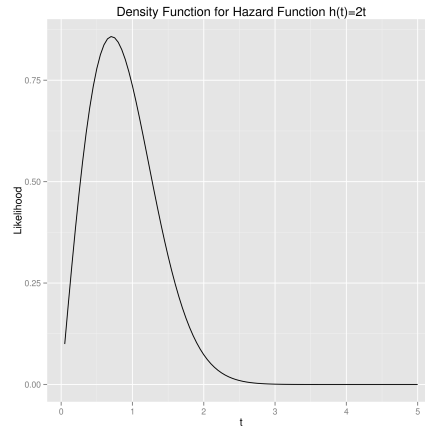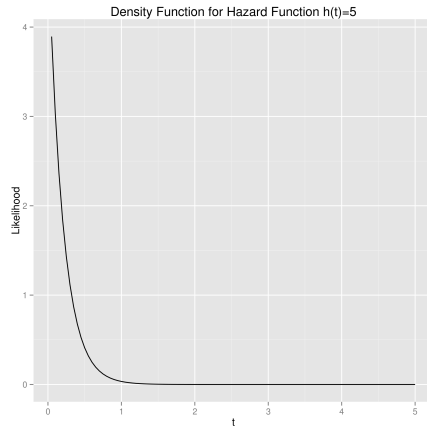$$f(t) = h(t) \cdot e^{-\int_0^t h(s)\,ds}$$

We looked at the following hazard functions to explore what their density would look like:

$$h(t) = 5$$
$$h(t) = 2t$$
$$h(t) = 4 - 2t$$
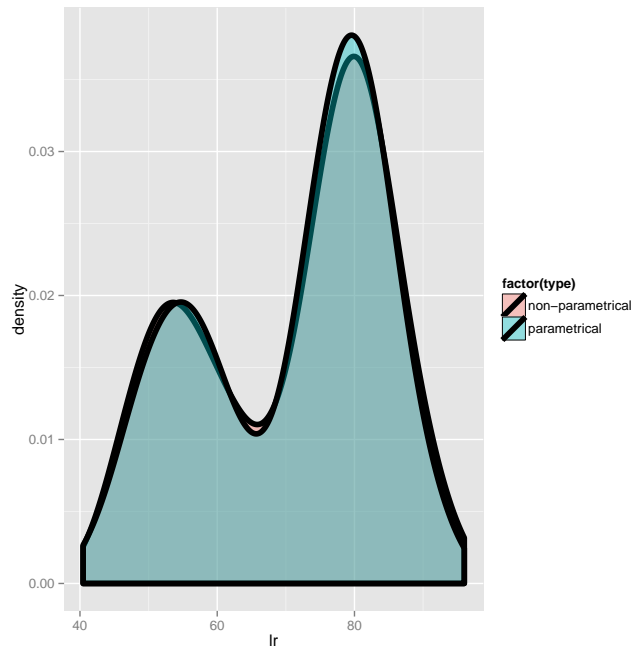$$h(t) = (t - 0.5)(t - 1)(t - 1.5)(t - 2) + 0.1$$

(Plots generated with `3.R`).

Problem 4.a-b

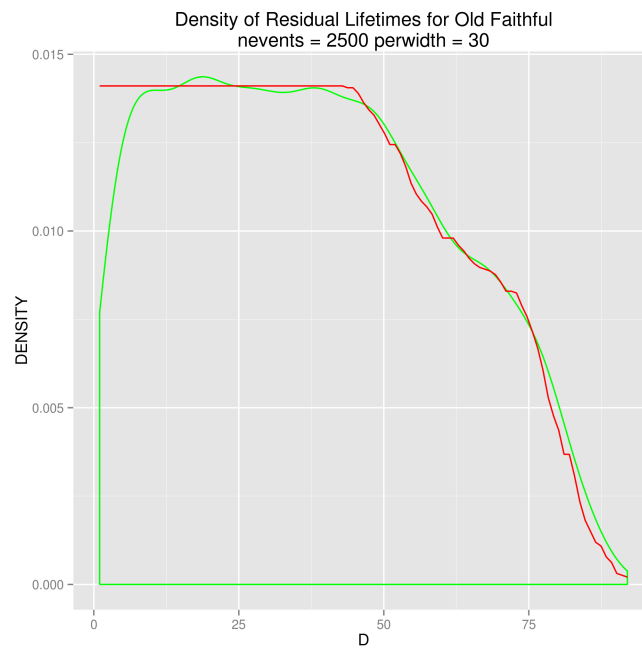See `4.R` for generating these plots.

Problem 4.c

```
1  simrenewal <- function(w, nevents, perwidth){
2    # Get nevents samples from w, with replacement
3    x <- sample(w, nevents, replace=TRUE)
4
5    # Get the running total of x, or the times at which a renewal occurs
6    times <- cumsum(x)
7
8    # Total number of samples
9    nsamp <- floor(times[length(times)] / perwidth)
10
11   # For each i from 0:nsamp-1, get the next time that occurace after i*perwidth,
12   # then subtract i*perwidth to get the residual for that sample.
13   resid <- sapply(0:(nsamp-1), function(x) (times[times>x*perwidth][1] - (x*perwidth)))
14
15   return(resid)
16 }
```

Problem 4.d-e

In the following plots, the red line results from eq. (11.21) and the green line is the simulated estimate. For 4f, we estimated the cdf of the Faithful data using the ecdf function, and then applied (11.21) as normal.

Density of Residual Lifetimes for Uniform Distribution
n = 300 nevents = 2500 perwidth = 1

Density of Residual Lifetimes for Old Faithful
nevents = 2500 perwidth = 30
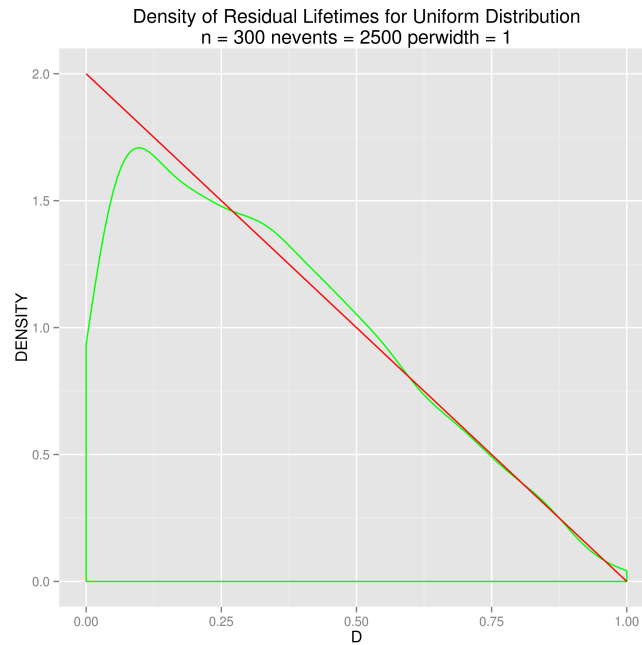
Problem 4.f

Creating the estimate using the function is simple. Our function outputs a sample of D, so to get an estimate of ED we simply take the mean of the sample outputted by our function. Doing this for simrenewal(10000, 20) gets us an estimated ED of 37.247.

The alternate estimate using Eq 11.31 is more complicated, and requires some derivation. Eq 11.31 is the following.

$$ED = \frac{E(L^2)}{2EL} \tag{1}$$

Where L is a random variable representing the lifetime. Now, we need to calculate EL and $E(L^2)$, which we will do using the results of the EM analysis. The results from EM give us a mixture distribution of two normals, where it selects from normal $N_1(\mu_1, \sigma_1)$ with probability p and $N_2(\mu_2, \sigma_2)$ with probability $p - 1$. First, EL, using the law of total expectation. Here, we will use an indicator variable B, which will indicate which normal our mixture selected. So we'll say it's 1 with probability p, and 2 with probability $1 - p$.

$$
\begin{aligned}
EL = E(L|B) \quad &= \begin{cases} E(N_1), w.p. \ p \\ E(N_2), w.p. \ 1 - p \end{cases} \\
&= E(N_1) * p + E(N_2) \\
&= mu_1 * p + mu_2 * p
\end{aligned} \tag{2}
$$

Since all of those values are parameters returned by the EM analysis, we're now done with EL. To find $E(L^2)$, we'll work from the variance of L using the rearrangement of (3.31) from the book

$$E(L^2) = Var(L) + (EL)^2 \tag{3}$$

We already have EL from earlier, so we just need the variance of L, which we'll do using the law of total variance with the same indicator variable B. To do this, we first need $E(L|B)$, which we already have, and $Var(L|B)$, which is quite similar to $E(L|B)$

$$
\begin{aligned}
Var(L|B) &= \begin{cases} Var(N_1), w.p. \ p \\ Var(N_2), w.p. \ 1 - p \end{cases} \\
&= \begin{cases} \sigma_1^2, w.p. \ p \\ \sigma_2^2, w.p. \ 1 - p \end{cases}
\end{aligned} \tag{4}
$$

Now we can continue with the law of total variance, and substitute into equation 4.3 from earlier.

$$
\begin{aligned}
E(L^2) &= E[Var(L|B)] + Var[E(L|B)] + EL^2 \\
&= \sigma_1^2 * p + \sigma_2^2 * (1 - p) + E(E(L|B)^2) - E(L|B)^2 + EL^2 \\
&= \sigma_1^2 * p + \sigma_2^2 * (1 - p) + mu_1^2 * p + mu_2^2 * (1 - p) - EL^2 + EL^2 \\
&= \sigma_1^2 * p + \sigma_2^2 * (1 - p) + mu_1^2 * p + mu_2^2 * (1 - p)
\end{aligned} \tag{5}
$$

Now we have $E(L^2)$ in the form of the results of the EM analysis, and we can finally calculate an ED estimate using eq (11.31), which with the paramters returned from EM analysis gave us $\boxed{36.747,}$ a reasonably close estimate to our simulations estimate.