Spread of a Random Sample from the Exponential

Hemant Banke (MD2107)

March 22, 2025

Statistical Computing 2 Assignment

1 Introduction to Problem

The aim of the assignment is to compare Edgeworth Approximation and saddle point approximations for the density function of Spread of Exponential Random Variables.

If $X_1, X_2, ..., X_{(n+1)}$ are n+1 independent, exponentially distributed random variables with common mean 1, then the spread $X_{(n+1)} - X_{(1)}$ has density

$$g(x) = n\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}e^{-kx}$$

This is also the density of the sum $S_n = Y_1 + ... + Y_n$ of independent, exponentially distributed random variables Y_j with respective means $1, 1/2, ..., 1/n$.

1.1 Proof of density

If $f_X(x)$ is the density of random variable X and $F_X(x)$ be its distribution function, then

$$f_X(x) = e^{-x} \quad ; x > 0$$

So the distribution function is,

$$\begin{aligned}
F_{X_{(n+1)}-X_{(1)}}(x) &= \int_0^\infty \bigcup_{i=1}^{n+1} P(t \le X_i \le t + x)dt \\
&= \int_0^\infty (e^{-t} - e^{-(t+x)})^{n+1}dt \\
&= \int_0^\infty e^{-t(n+1)} + \sum_{k=1}^{n+1}(-1)^k\binom{n+1}{k}e^{-t(n+1)}e^{-xk}dt \\
&= \left(1 + \sum_{k=1}^{n+1}(-1)^k\binom{n+1}{k}e^{-xk}\right)\int_0^\infty e^{-t(n+1)}dt \\
&= \frac{1}{n+1}\left(1 + \sum_{k=1}^{n+1}(-1)^k\binom{n+1}{k}e^{-xk}\right)
\end{aligned}$$

Hence, the density

$$\begin{aligned}
f_{X_{(n+1)}-X_{(1)}}(x) &= \frac{d}{dx}\frac{1}{n+1}\left(1 + \sum_{k=1}^{n+1}(-1)^k\binom{n+1}{k}e^{-xk}\right) \\
&= \sum_{k=1}^{n+1}(-1)^k\frac{1}{n+1}\binom{n+1}{k}e^{-xk}(-k) \\
&= n\sum_{k=1}^{n}(-1)^{k-1}\binom{n-1}{k-1}e^{-xk}
\end{aligned}$$

Also as, $Y_j \sim exp(1/j)$, $f_{Y_j}(x) = je^{-jx}$.
So its characteristic function,

$$\hat{f}_{Y_j}(y) = \int_0^\infty e^{iyx}je^{-jx}dx = \frac{j}{j - iy}$$

Hence,

$$\hat{f}_{Y_1+...+Y_n}(y) = \prod_{j=1}^{n}\frac{j}{j - iy}$$

Expressing in form of sums will help us in finding relevant Edgeworth Approximation and saddle point approximations for the density function $g(x)$ (density function of $Y_1 + Y_2 + ... + Y_n$)

2 Edgeworth Approximation

Let, κ_j be the j'th cumulant of Y's and σ^2 be a common variance, then define $\rho_j = \kappa_j/\sigma^j$. Then the characteristic function of $T_n = (S_n - E(S_n))/\sqrt{Var(S_n)}$ is,

$$e^{-\frac{y^2}{2}}\left[1 + \frac{\rho_3(iy)^3}{6\sqrt{n}} + \frac{\rho_4(iy)^4}{24n} + + \frac{\rho_3^2(iy)^6}{72n} + O(n^{-3/2})\right]$$

Fourier inversion of this leads us to asymptotic expansion for the density of T_n ,

$$f_{T_n}(x) = \phi(x)\left[1 + \frac{\rho_3 H_3(x)}{6\sqrt{n}} + \frac{\rho_4 H_4(x)}{24n} + \frac{\rho_3^2 H_6(x)}{72n} + O(n^{-3/2})\right] \quad (1)$$

where $H_n(x)$ is the Hermite polynomial.

Here, we will use the fact that the cumulant generating function of S_n is

$$nK(t) = \sum_{j=1}^{n} ln(M_{Y_j}(t)) = \sum_{j=1}^{n} ln(\frac{j}{j-t}) = -\sum_{j=1}^{n} ln(1 - t/j)$$

So, the k'th cumulant becomes,

$$nK^{(k)}(0) = \sum_{j=1}^{n} \frac{d^k}{dt^k_{t=0}} ln(\frac{j}{j-t}) = (k-1)! \sum_{j=1}^{n} \frac{1}{j^k}$$

This leads to

$$\rho_k = \rho_k(0) = \frac{nK^{(k)}(0)}{[nK^{(2)}(0)]^{k/2}} = (k-1)! \frac{\sum_{j=1}^{n} \frac{1}{j^k}}{\left(\sum_{j=1}^{n} \frac{1}{j^2}\right)^{k/2}}$$

Hence, we can estimate the density $g(x)$ of S_n by evaluating equation (1) as,

$$g(x_0) = f_{T_n}\left(\left[x_0 - nK^{(1)}(0)\right]/\sqrt{nK^{(2)}(0)}\right)/\sqrt{nK^{(2)}(0)} \quad (2)$$

3 Saddle Point Approximations

For Saddle Point Approximations we tilt the density $g(x)$ to density $e^{tx - nK(t)}g(x)$, which has k'th cumulant

$$nK^{(k)}(t) = (k-1)! \sum_{j=1}^{n} \frac{1}{(j-t)^k}$$

We can achieve an arbitrary mean x_0 by choosing t_0 to be the solution of the equation $nK^{(1)}(t_0) = x_0$ (solved numerically), i.e.

$$nK^{(1)}(t_0) = \sum_{j=1}^{n} \frac{1}{(j-t_0)} = x_0 \quad (3)$$

Once t_0 is chosen, the standardized tilted density approximated using (1), giving us :

$$g(x_0) = \frac{e^{-t_0 x_0 + nK(t_0)}}{\sqrt{2\pi nK^{(2)}(t_0)}}\left[1 + O(n^{-1})\right] \quad (4)$$

Further terms can be included in the saddle point approximation if we substitute the appropriate normalized cumulants

$$\rho_k(t_0) = \frac{nK^{(k)}(t_0)}{[nK^{(2)}(t_0)]^{k/2}} = (k-1)! \frac{\sum_{j=1}^{n} \frac{1}{(j-t_0)^k}}{\left(\sum_{j=1}^{n} \frac{1}{(j-t_0)^2}\right)^{k/2}}$$

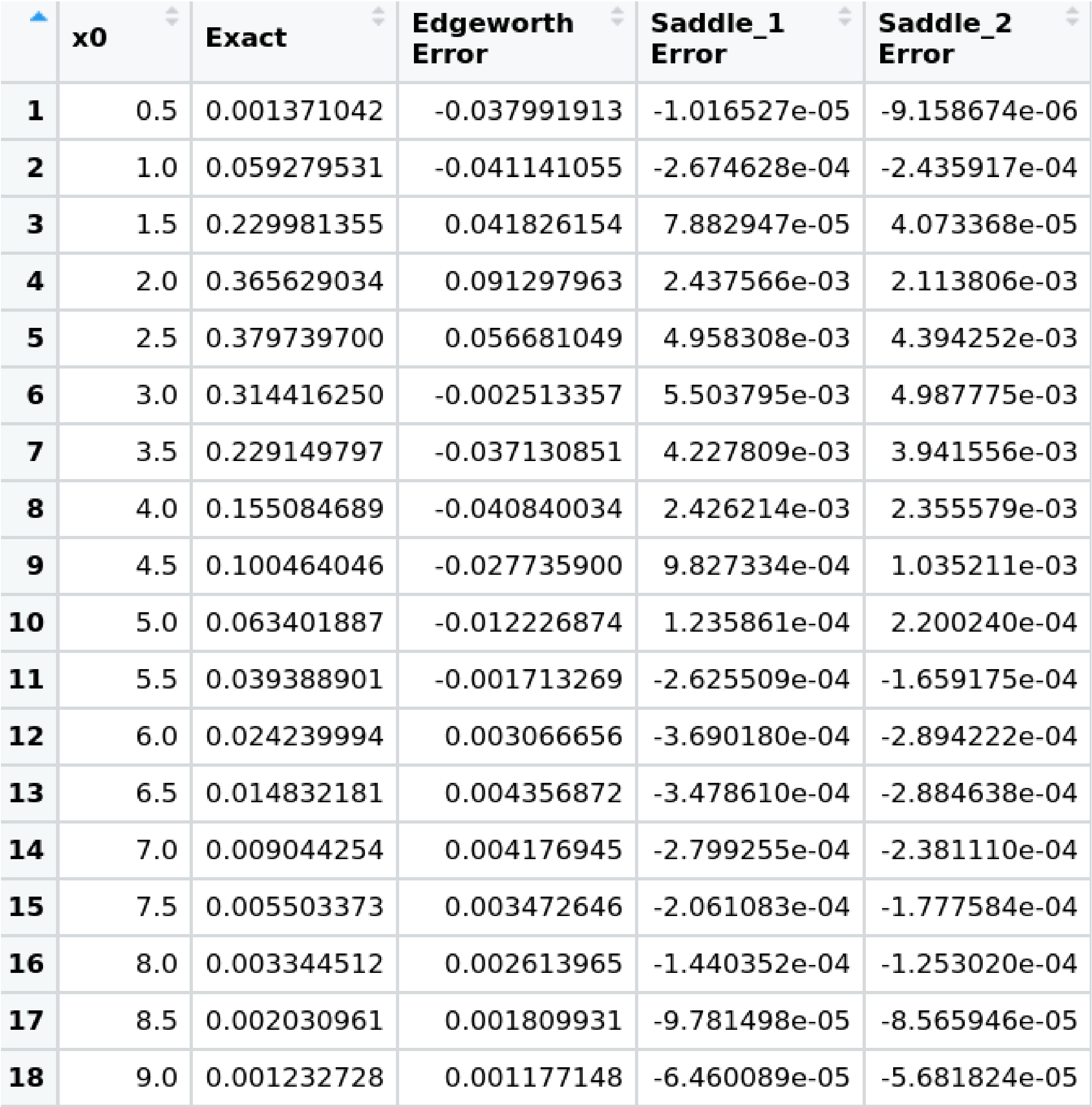
of the tilted density in the Edgeworth expansion (2), giving us

$$g(x_0) = \frac{e^{-t_0 x_0 + nK(t_0)}}{\sqrt{2\pi nK^{(2)}(t_0)}}\left[1 + \frac{3\rho_4(t_0) - 5\rho_3^2(t_0)}{24n} + O(n^{-3/2})\right] \quad (5)$$

4 Numerical Analysis

To solve the numerical equation (3), we use the *'rootSolve::uniroot()'* method in R to get a root of the equation.

Now we use equation (2), (4) and (5) to approximate the density using Edgeworth and the two saddle point approximations, respectively, giving us the following errors for $n = 10$ and the listed x_0 values,



	x0	Exact	Edgeworth Error	Saddle_1 Error	Saddle_2 Error
1	0.5	0.001371042	-0.037991913	-1.016527e-05	-9.158674e-06
2	1.0	0.059279531	-0.041141055	-2.674628e-04	-2.435917e-04
3	1.5	0.229981355	0.041826154	7.882947e-05	4.073368e-05
4	2.0	0.365629034	0.091297963	2.437566e-03	2.113806e-03
5	2.5	0.379739700	0.056681049	4.958308e-03	4.394252e-03
6	3.0	0.314416250	-0.002513357	5.503795e-03	4.987775e-03
7	3.5	0.229149797	-0.037130851	4.227809e-03	3.941556e-03
8	4.0	0.155084689	-0.040840034	2.426214e-03	2.355579e-03
9	4.5	0.100464046	-0.027735900	9.827334e-04	1.035211e-03
10	5.0	0.063401887	-0.012226874	1.235861e-04	2.200240e-04
11	5.5	0.039388901	-0.001713269	-2.625509e-04	-1.659175e-04
12	6.0	0.024239994	0.003066656	-3.690180e-04	-2.894222e-04
13	6.5	0.014832181	0.004356872	-3.478610e-04	-2.884638e-04
14	7.0	0.009044254	0.004176945	-2.799255e-04	-2.381110e-04
15	7.5	0.005503373	0.003472646	-2.061083e-04	-1.777584e-04
16	8.0	0.003344512	0.002613965	-1.440352e-04	-1.253020e-04
17	8.5	0.002030961	0.001809931	-9.781498e-05	-8.565946e-05
18	9.0	0.001232728	0.001177148	-6.460089e-05	-5.681824e-05

Figure 1: Table displays the exact density and the errors (exact values minus approximate values) committed in using the Edgeworth expansion (2) and the two saddle point expansions (4) and (5) when n = 10

We notice that both saddle point expansions clearly outperform the Edgeworth expansion except very close to the mean $\sum_{j=1}^{n} 1/j = 2.93$. Saddle point expansions perform very well even when this setup is far from the ideal of a sum of i.i.d. random variables.

The refined saddle point expansion (5) is an improvement over the ordinary saddle point expansion (4) in the tails of the density but not necessarily in the center, which is why (5) has similar errors to (4) around mean but lower errors away from the mean.

This can be better observed from the following graphs,

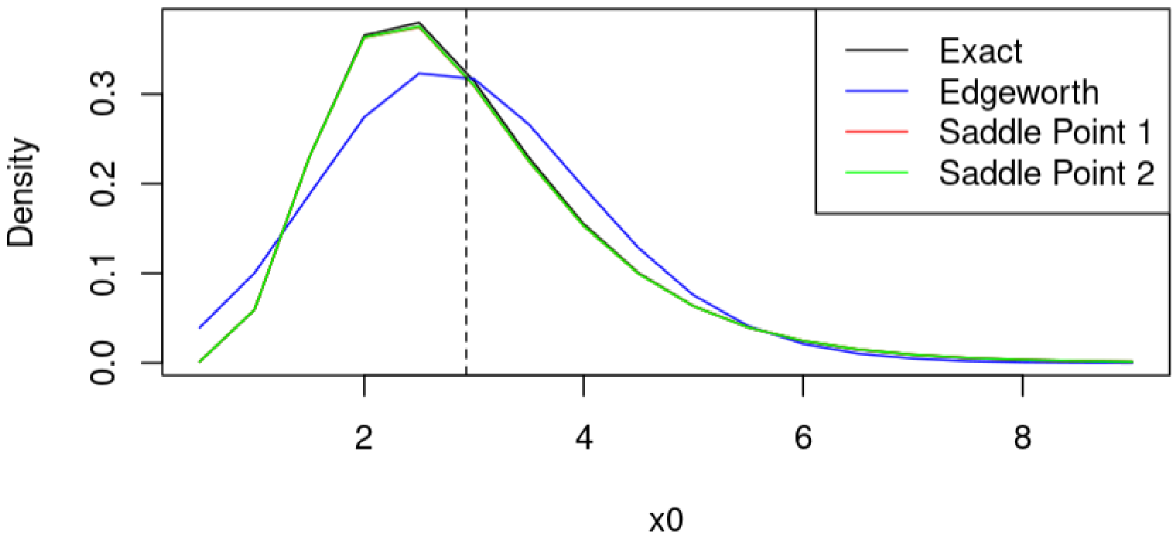


Figure 2: The Exact Density compared with the three density approximations

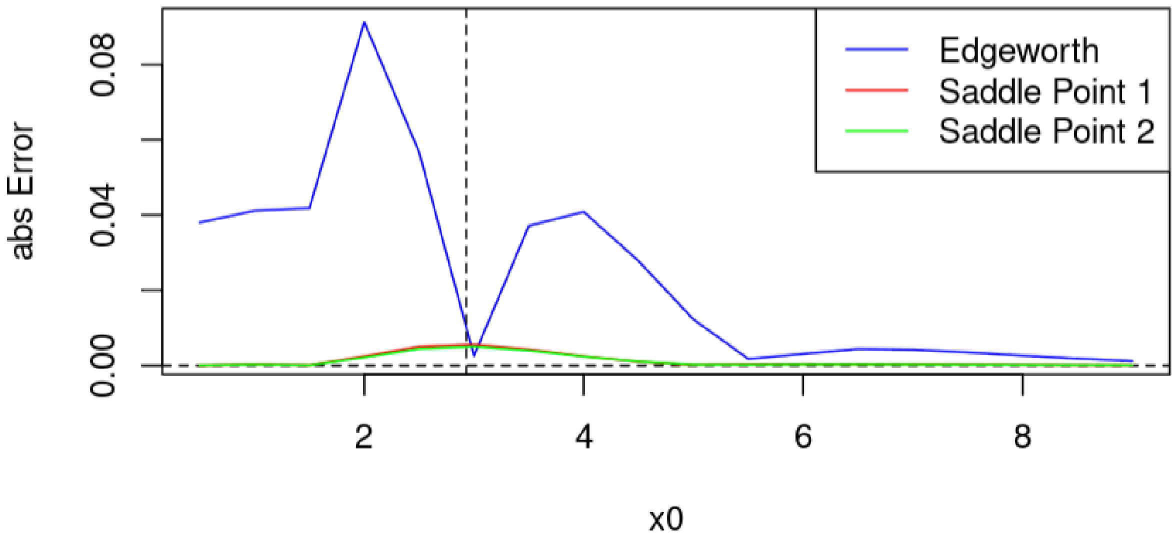


Figure 3: The Absolute Error of the three density approximations

Clearly, Saddle point approximations perform way better than Edgeworth expansions with (5) performing better than (4). But close to the mean, saddle point approximations perform worse than Edgeworth.

5 R Code

```r
library(EQL)
library(rootSolve)

n = 10
nK = function(t){
  return (sum(sapply(1:n, function (j) log(j/(j - t)))))
}
cumulant = function(k, t){
  return (factorial(k-1) * sum(sapply(1:n, function (j) 1/(j - t)^k)))
}
rho = function(k, t){
  return (cumulant(k, t)/(cumulant(2, t))^(k/2))
}

## Exact density
g = function(x0){
  return (n * sum(sapply(1:n, function (k) {
    (-1)^(k-1) * choose(n-1, k-1) * exp(-k*x0)
  })))
}

## Edgeworth Approximation
f_Tn = function(x){
  return (dnorm(x)*(1 + (rho(3, 0)*hermite(x, 3))/(6*sqrt(n))
                    + (rho(4, 0)*hermite(x, 4))/(24*n)
                    + ((rho(3, 0)^2)*hermite(x, 6))/(72*n) ))
}

edgeworth = function(x0){
  return (f_Tn((x0 - cumulant(1, 0))/sqrt(cumulant(2, 0)))/sqrt(cumulant(2, 0)) )
}

## Finding t0
t0_equation = function(t0, x0){
  return (cumulant(1, t0) - x0)
}
get_t0 = function(x0){
  interval = c(0,1)
  if (x0 < 3){
    interval = c(-15, 0)
  }
  return (uniroot(function(t0) t0_equation(t0, x0), interval)$root)
}

## Saddle Point 1
sp1 = function(x0, t0){
  return (exp(-t0*x0 + nK(t0))/sqrt(2*pi*cumulant(2, t0)))
}

## Saddle Point 2
sp2 = function(x0, t0){
  return ((exp(-t0*x0 + nK(t0))/sqrt(2*pi*cumulant(2, t0)))
          *(1 + (3*rho(4,t0) - 5*rho(3, t0)^2)/(24*n)) )
}
```

```r
dt = data.frame(x0 = seq(0.5,9,by=0.5))
# Exact
dt[, 'Exact'] = sapply(dt[, 'x0'], function(x0) g(x0))
# Edgeworth
dt[, 'Edgeworth'] = sapply(dt[, 'x0'], function(x0) edgeworth(x0))
dt[, 'Edgeworth Error'] = dt[, 'Exact'] - dt[, 'Edgeworth']
# Get t0
dt[, 't0'] = sapply(dt[, 'x0'], function(x0) get_t0(x0))
# Saddle Point 1
dt[, 'Saddle_1'] = sapply(1:nrow(dt), function(i) sp1(dt[i, 'x0'], dt[i, 't0']))
dt[, 'Saddle_1 Error'] = dt[, 'Exact'] - dt[, 'Saddle_1']
# Saddle Point 2
dt[, 'Saddle_2'] = sapply(1:nrow(dt), function(i) sp2(dt[i, 'x0'], dt[i, 't0']))
dt[, 'Saddle_2 Error'] = dt[, 'Exact'] - dt[, 'Saddle_2']
```