The gamma distribution

The gamma distribution is important in fields such as econometrics and life modelling. Parameterised by the rate, θ, and the shape, α, the distribution function is

$$F(x) = \frac{\gamma(\alpha, \theta x)}{\Gamma(\alpha)},$$

where γ is the lower incomplete gamma function, while the probability density function is

$$f(x|\theta, \alpha) = \frac{\theta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\theta x}.$$

This distribution has mean $\frac{\alpha}{\theta}$ and variance $\frac{\alpha}{\theta^2}$.

The inverse of F doesn't have a generic closed form, but for particular values of α it can. In particular, for $\alpha = 1$, the gamma distribution matches the exponential distribution, while for $\alpha = 2$, the inverse has the form

$$F^{-1}(x) = -\frac{W_p(\frac{x-1}{e}) + 1}{\theta},$$

where W_p is the principle branch of the Lambert W function.

Programs

We use the implementation of the incomplete gamma function found in the package `expint`, and the implementation of the Lambert W function found in the package `pracma`.

The gamma pdf is implemented as `gamma_pdf`, the distribution function as `gamma_distribution`, and the inverse of the distribution function (for the special case $\alpha = 2$) as `gamma_distribution_special_inv`.

We can sample from the gamma distribution with the following code:

```
theta <- 2.2
alpha <- 2
n <- 200
distribution <- Curry(gamma_distribution_special_inv, rate = theta)

gamma_samples <- distribution_sampler(distribution, n)
head(gamma_samples)
```

```
## [1] 0.6385382 0.5945956 0.5406885 1.7160053 1.0651728 0.2929526
```

R also has a built-in exponential distribution with pdf `dgamma`, which can be sampled with `rgamma`. We can test that this matches our implementation using a one-sample Kolmogorov-Smirnov test:

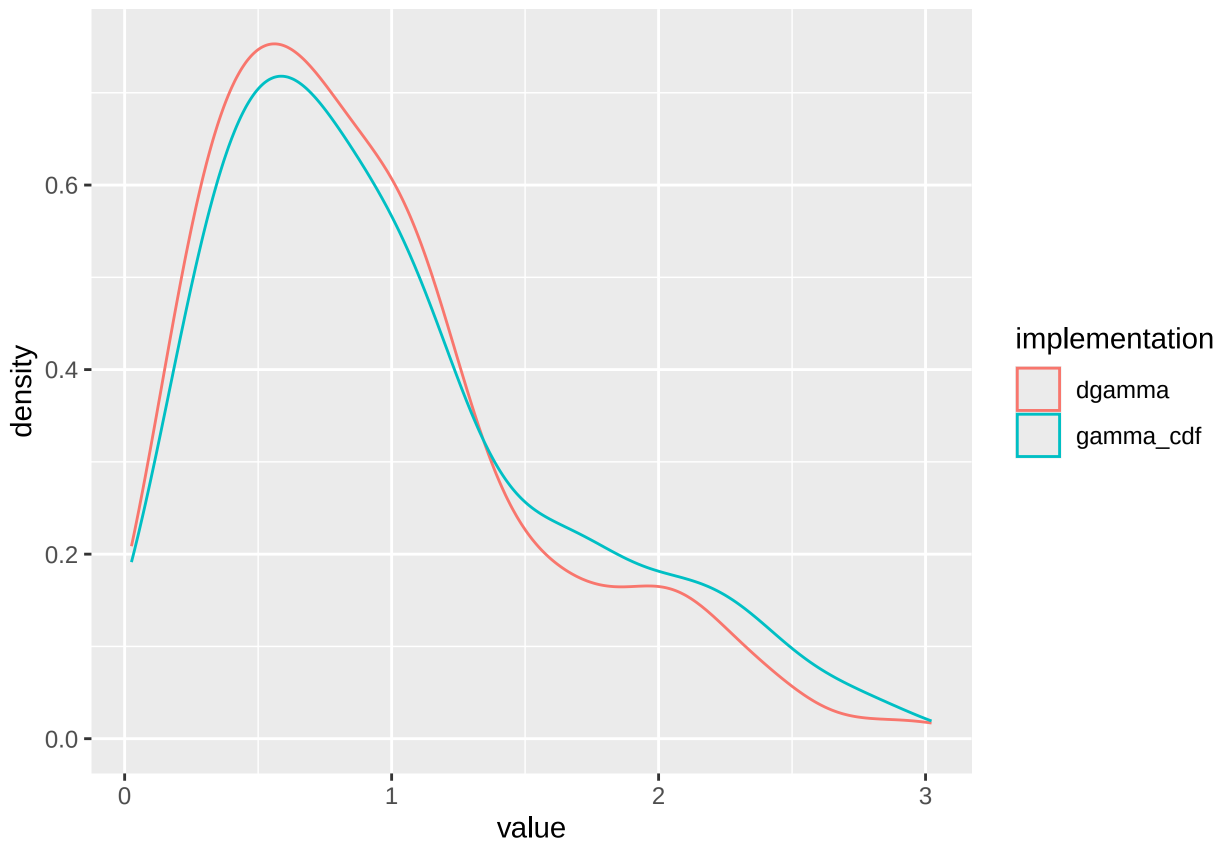
```
ks_results <- ks.test(gamma_samples, "dgamma", shape = alpha, rate = theta)
ks_results[["p.value"]]
```

```
## [1] 3.213735e-166
```

In this instance, since $p < 0.05$, we conclude that the distributions match.

We can also plot the distributions against each other:

```
dat <- data.frame(implementation =
                    factor(rep(c("gamma_cdf", "dgamma"), each = n)),
                  value = c(gamma_samples,
                            rgamma(n, rate = theta, shape = alpha)))

ggplot(dat, aes(x = value, colour = implementation)) + geom_density()
```



Here, we have some noise from our relatively low number of samples, but the distributions clearly have the same shape.

Problems

Problem 5

Take $f(x|\theta) = \theta^2 x e^{-\theta x}$, $x > 0$, and integrate it to find $F(x)$. Can you compute F^{-1} in closed form?

Solution We can compute

$$F(x) = \int_0^x f(t|\theta)\ \mathrm{dt}$$

$$
\begin{aligned}
&= [-\theta t e^{-\theta t}]_{t=0}^{x} + \int_0^x \theta e^{-\theta t}\ \mathrm{dt} \\
&= -\theta x e^{-\theta x} - [e^{-\theta t}]_{t=0}^{x} \\
&= -\theta x e^{-\theta x} + 1 - e^{-\theta x} \\
&= 1 - (\theta x + 1)e^{-\theta x}.
\end{aligned}
$$

Then we can find $F^{-1}(x)$ as follows:

$$
\begin{aligned}
& x = 1 - (\theta F^{-1}(x) + 1)e^{-\theta F^{-1}(x)} \\
\implies & x - 1 = e \times -(\theta F^{-1}(x) + 1)e^{-(\theta F^{-1}(x)+1)} \\
\implies & W_p(\frac{x-1}{e}) = -(\theta F^{-1}(x) + 1) \\
\implies & \frac{-W_p(\frac{x-1}{e}) - 1}{\theta} = F^{-1}(x),
\end{aligned}
$$

where W_p is the principle branch of the Lambert W function.

Problem 6

The log-likelihood function is now

$$\ell(\theta) = \ln \prod_{i=1}^{n} f(x_i|\theta).$$

Calculate the maximum likelihood estimator for θ.

Solution We can compute

$$
\begin{aligned}
\ell(\theta) &= \ln(\theta^{2n}(\prod x_i)e^{-\theta(\sum x_i)}) \\
&= 2n\ln(\theta) + \sum(\ln x_i) - \theta \sum x_i.
\end{aligned}
$$

To find the maximum likelihood estimator, we can solve

$$0 = \ell'(\theta) = n(\frac{2}{\theta} - \bar{x}),$$

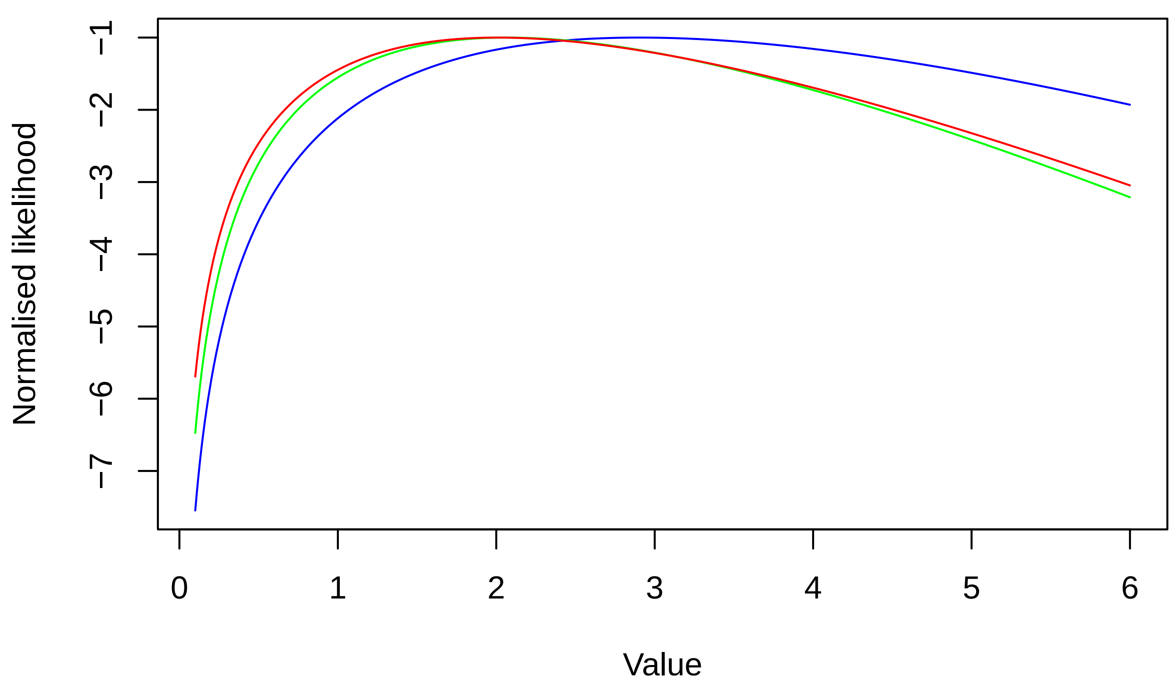
where $\bar{x}$ is the mean of the x_i. So $\hat{\theta} = \frac{2}{\bar{x}}$.

Problem 7

Take $\theta_0 = 2.2$, generate a random sample of $x_1, ..., x_n$ from $f(x|\theta_0)$, and plot $\ell(\theta)$ against θ, for $n = 10, 30, 50$. For each sample, calculate the maximum likelihood estimator for θ, and compare it with θ_0, describing any similarities or differences between this case and that in Problem 3.

Solution We plot ℓ with the following code:

```r
log_likelihood <- function(n, samples, rate) {
  n * 2 * log(rate) + sum(sapply(samples, log)) - rate * sum(samples)
}

gamma_samples_1 <- distribution_sampler(distribution, 10)
gamma_samples_2 <- distribution_sampler(distribution, 30)
gamma_samples_3 <- distribution_sampler(distribution, 50)

el_1 <- Curry(log_likelihood, n = 10, samples = gamma_samples_1)
el_2 <- Curry(log_likelihood, n = 30, samples = gamma_samples_2)
el_3 <- Curry(log_likelihood, n = 50, samples = gamma_samples_3)

theta_1 <- 2 / (sum(gamma_samples_1) / 10)
theta_2 <- 2 / (sum(gamma_samples_2) / 30)
theta_3 <- 2 / (sum(gamma_samples_3) / 50)

x <- seq(0.1, 6, 0.01)
plot(x, sapply(x, el_1) / -el_1(theta_1), type = "l", col = "blue",
     xlab = "Value", ylab = "Normalised likelihood")
lines(x, sapply(x, el_2) / -el_2(theta_2), col = "green")
lines(x, sapply(x, el_3) / -el_3(theta_3), col = "red")
```



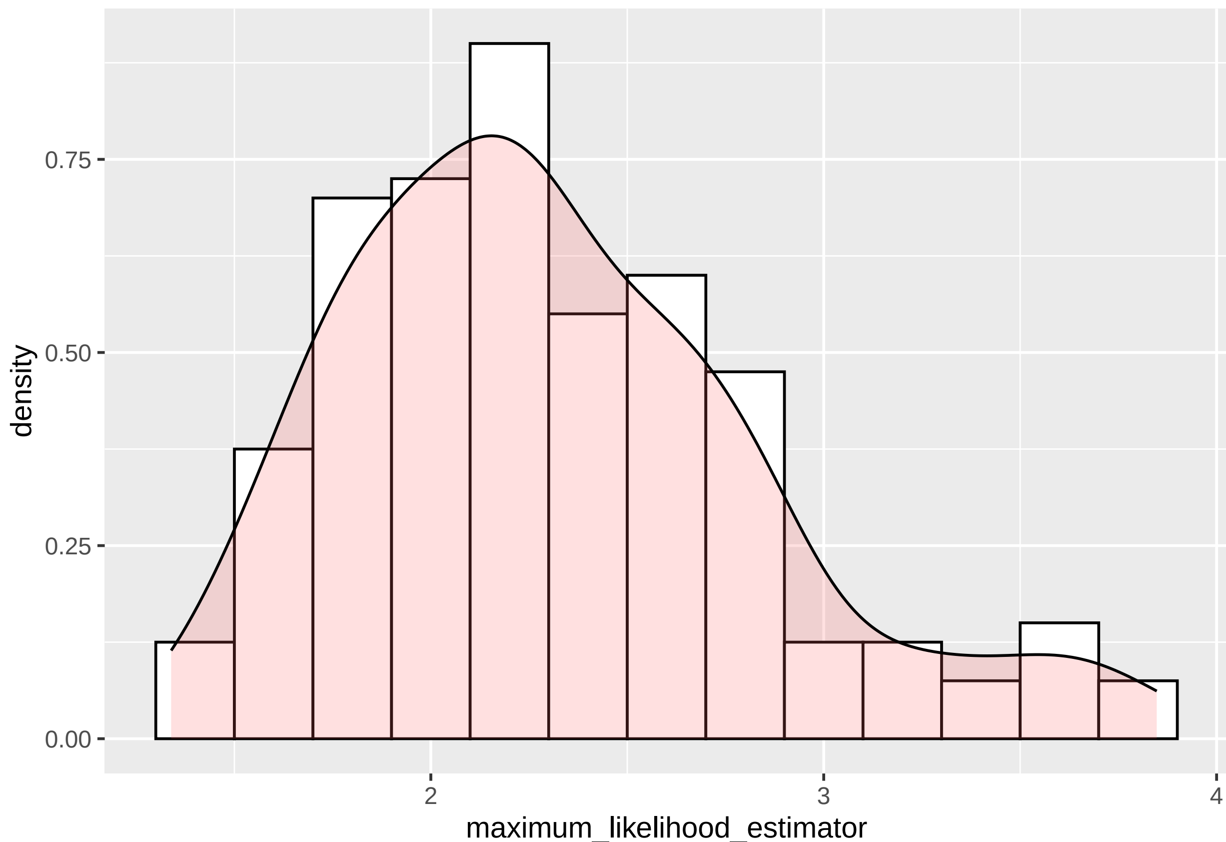
We observe a similar phenomenon to Problem 3: all three plots are similar, but smaller sample sizes result in poorer estimations of the median, and different gradients for ℓ.

Problem 8

We investigate the distribution of $\hat{\theta}$ as follows. Take $\theta_0 = 2.2$ and $N = 200$. Take $x(1), ..., x(N)$ as N independent random samples each of size $n = 10$ from $f(x|\theta_0)$. Let $\hat{\theta}(1), ..., \hat{\theta}(N)$ be the corresponding maximum likelihood estimators of θ. Generate the histogram of $\hat{\theta}(1), ..., \hat{\theta}(N)$. How does this histogram change if we increase n from 10 to 50?

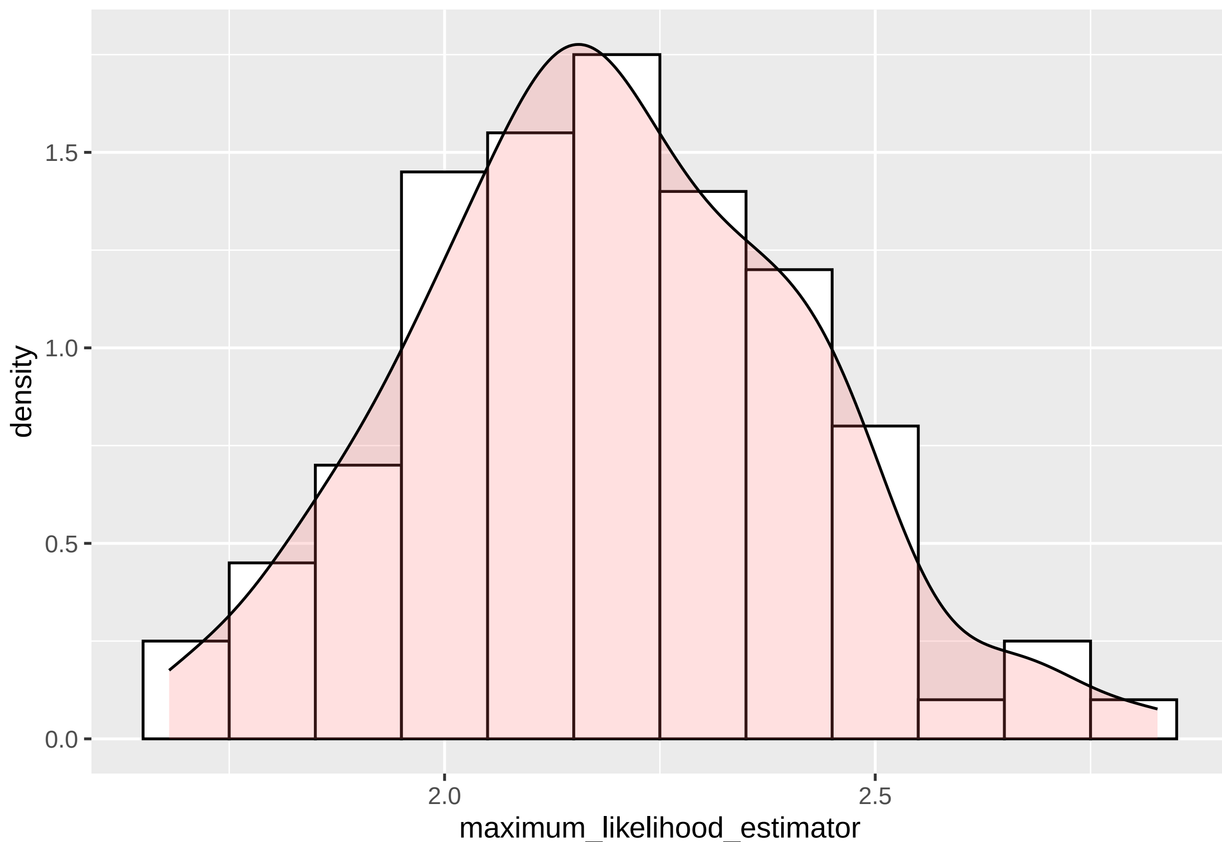
Solution We generate the histogram with the following code:

```
plot_mle_histogram <- function(n, N, bin_width) {
  mle_theta <- list()
  for (i in 1: N) {
    mle_theta[[i]] <- 2 * n / sum(distribution_sampler(distribution, n))
  }
  mle_data <- data.frame(maximum_likelihood_estimator = as.numeric(mle_theta))

  ggplot(mle_data, aes(x = maximum_likelihood_estimator)) +
    geom_histogram(aes(y = after_stat(density)),
                   binwidth = bin_width,
                   colour = "black", fill = "white") +
    geom_density(alpha = .2, fill = "#FF6666")
}

plot_mle_histogram(10, 200, 0.2)
```



Now, if we increase n to 50, we get the following histogram:

```r
plot_mle_histogram(50, 200, 0.1)
```



Notice that the maximum likelihood estimator now has significantly less variance. This makes sense: as our number of samples increases, we expect estimators closer to the true value.