Introduction to Bayesian Statistics with R

3: Exercise solutions

Jack Kuipers

28 November 2022

First we load the tidyverse and set a seed.

```
library(tidyverse); set.seed(42)
```

Bonus Exercise 3.1 - Monte Carlo integration

Computing expectations can be applied to any continuous function

$$E[f(x)] = \int f(x)p(x)\mathrm{d}x$$

so that integrals where we recognise $p(x)$ as (proportional to) a probability distribution may be estimated with Monte Carlo methods since

$$E[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

for N random samples x_i sampled according to $p(x)$. Use samples from a Gaussian to estimate the following three integrals:

$$\int |x|\mathrm{e}^{-x^2}\mathrm{d}x\,, \qquad \int \sin(x)\mathrm{e}^{-x^2}\mathrm{d}x\,, \qquad \int \cos(x)\mathrm{e}^{-x^2}\mathrm{d}x$$

Compare the estimated values to the exact values of the integrals. How does the accuracy depend on the sample size?

Looking at the three integrals we can identify a normal distribution. The Gaussian has the following general form:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}\,.$$

For our problems we thus need to multiply $\sqrt{\pi}$ and correctly match the respective mean and variance:

$$\mathrm{e}^{-x^2} = \sqrt{\pi}\frac{1}{\sqrt{2\pi\frac{1}{2}}}\mathrm{e}^{-\frac{(x-0)^2}{2\frac{1}{2}}} \tag{1}$$

to the values $\mu = 0$ and $\sigma^2 = \frac{1}{2}$. Thus we sample N particles with this parameterisation:

```
N <- 1e5
normal_samples <- rnorm(N, mean = 0, sd = sqrt(1/2))
```

Then we evaluate the samples with the three functions above (don't forget to multiply the constant factor $\sqrt{\pi}$) and average them. That's it.

```
sqrt(pi)*mean(abs(normal_samples))
```

```
## [1] 1.003612
```

```
sqrt(pi)*mean(sin(normal_samples))
```

```
## [1] -0.003779498
```

```
sqrt(pi)*mean(cos(normal_samples))
```

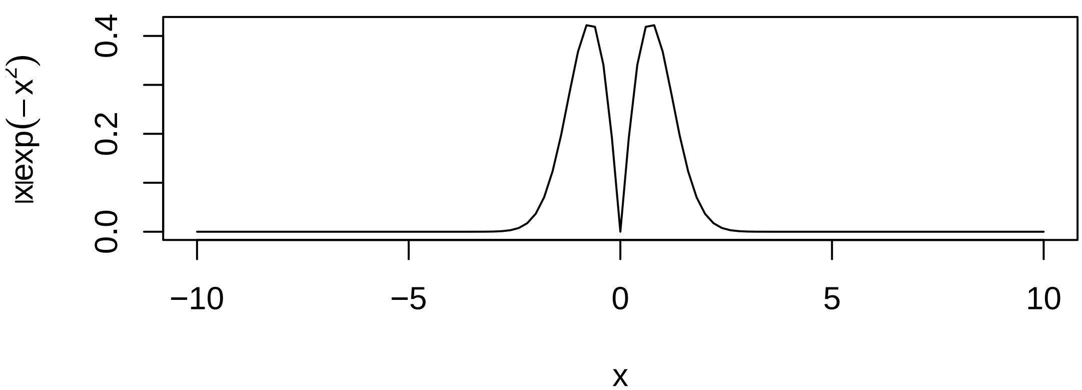
```
## [1] 1.377853
```

For the comparison to the real values, you can integrate analytically or use `R`'s `integrate` function. The errors from the true value, like standard errors in general, decrease like $\frac{1}{\sqrt{N}}$.

Absolute

This integral can be evaluated by using the symmetry around the y-axis and the fact that $\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{e}^{-x^2} = -2x\mathrm{e}^{-x^2}$:

```
curve(abs(x)*exp(-x^2), from = -10, to = 10, ylab = expression(abs(x)*exp(-x^2)))
```



Integrated with `R`:

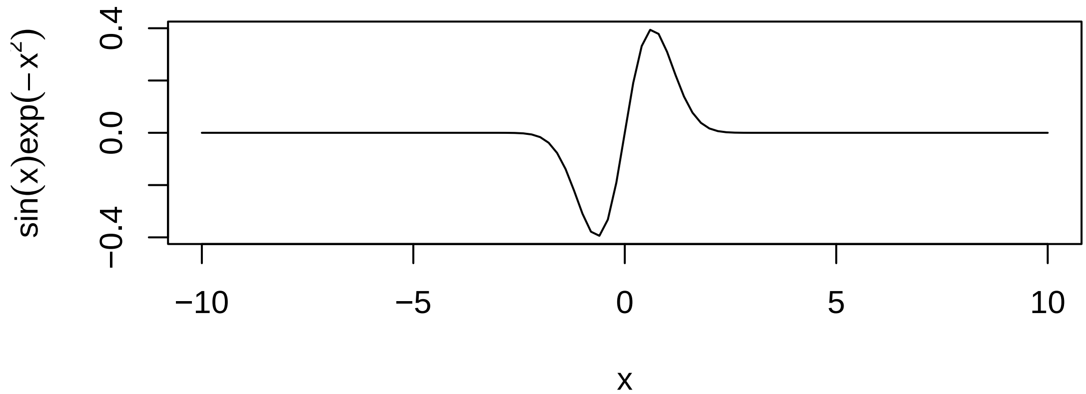
```
integrate(function(x) abs(x)*exp(-x^2), -Inf, Inf)$value
```

```
## [1] 1
```

Sine

Here it suffices to look at the symmetry along the axis:

```
curve(sin(x)*exp(-x^2), from = -10, to = 10, ylab = expression(sin(x)*exp(-x^2)))
```



Integrated with R:

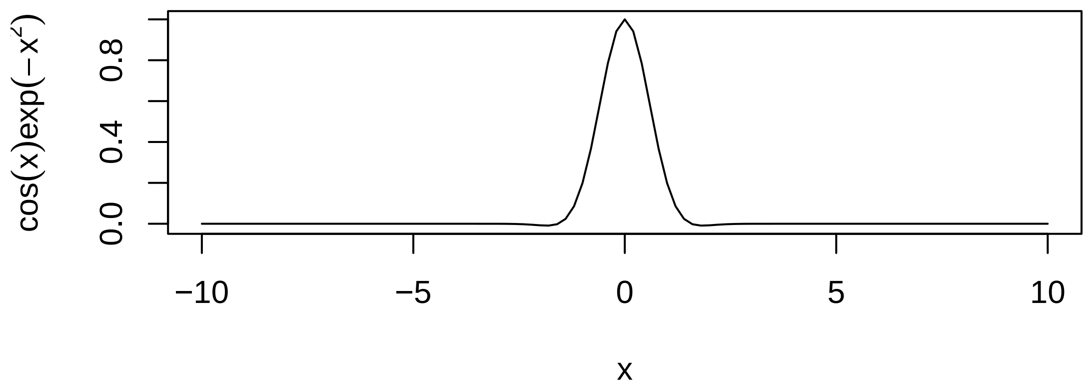
```
integrate(function(x) sin(x)*exp(-x^2), -Inf, Inf)$value
```

```
## [1] 0
```

Cosine

Estimating this integral is somewhat harder:

```
curve(cos(x)*exp(-x^2), from = -10, to = 10, ylab = expression(cos(x)*exp(-x^2)))
```



Integrated with R:

```
integrate(function(x) cos(x)*exp(-x^2), -Inf, Inf)$value
```

```
## [1] 1.380388
```

The exact value of the integral is $\sqrt{\pi}\exp(-\frac{1}{4})$: 1.3803884.

Exercise 3.2 - MCMC

For MCMC we can walk randomly and accept according the the MH ratio to eventually sample proportionally to any target distribution $p(x)$. For example, a Student-t distribution with $\nu = 5$.

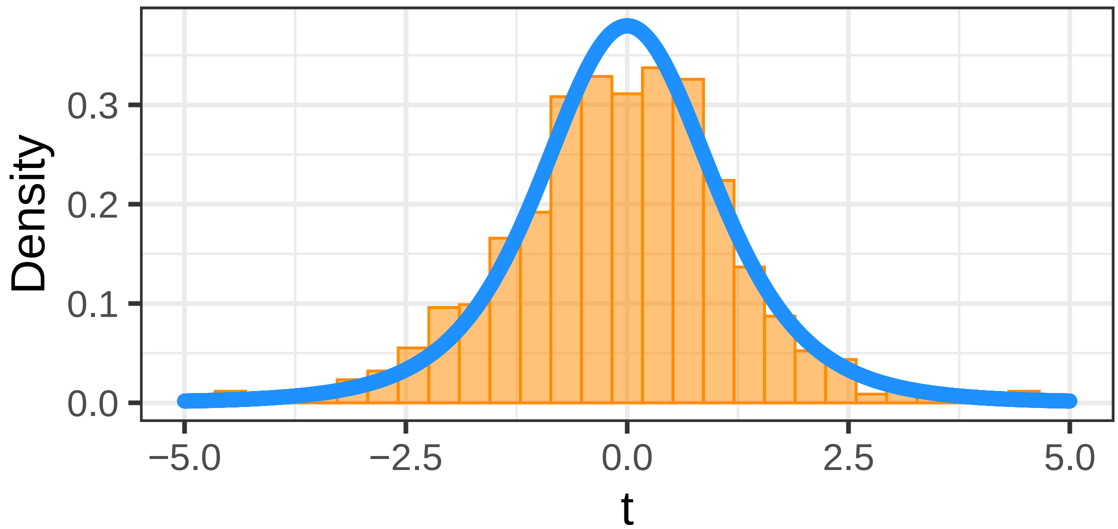
Examine the output MCMC chain for different lengths. How many samples would we need to get close to the Student-t distribution?

Use the samples to estimate (see description in Bonus Exercise 3.1)

$$\int \cos(t) f_5(t) \mathrm{d}t\,, \qquad f_\nu(t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

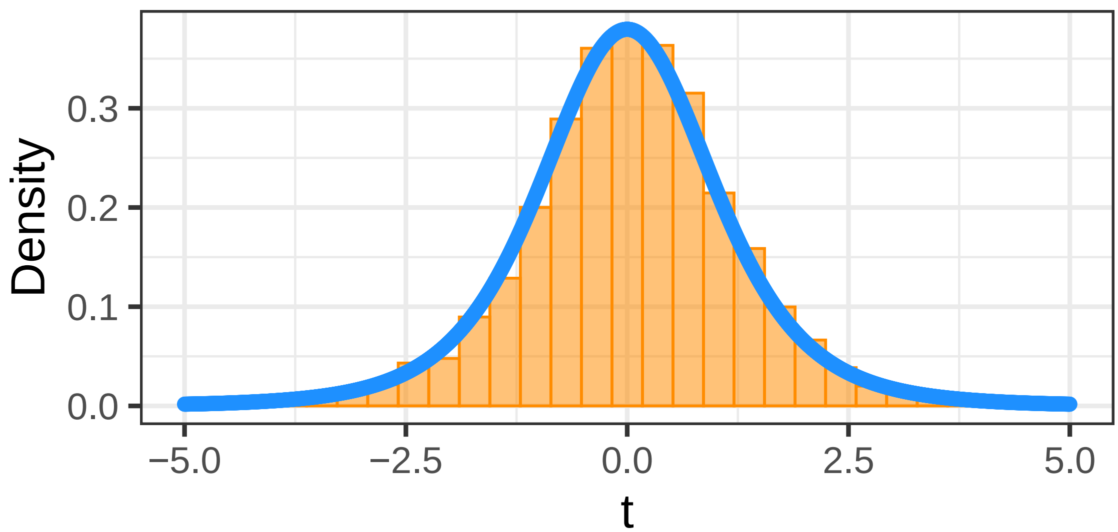
First we run a short chain with the default length of 1000 iterations:

```
short_chain <- basicMCMC(nu = 5)
```

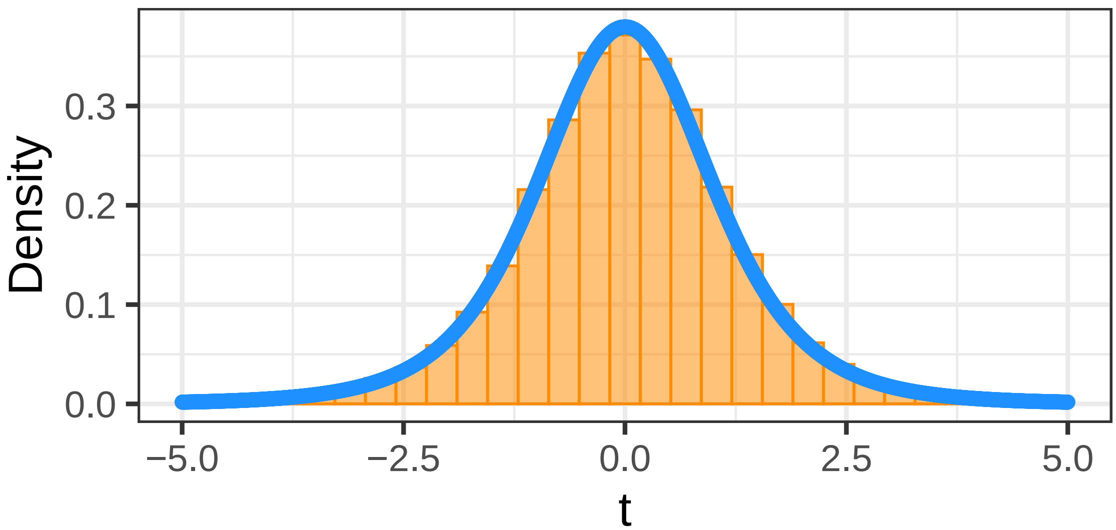


It's not so bad, but a few discrepancies. Let's try longer chains

```
longer_chain <- basicMCMC(n_its = 1e4, nu = 5)
```




```
even_longer_chain <- basicMCMC(n_its = 1e5, nu = 5)
```



and they start to look quite good.

For the integral we simply evaluate the cosine at each of our sampled values and get

```
mean(cos(short_chain))
```

```
## [1] 0.4860796
```

```
mean(cos(longer_chain))
```

```
## [1] 0.5329474
```

```
mean(cos(even_longer_chain))
```

```
## [1] 0.5209446
```

```
very_long_chain <- basicMCMC(n_its = 1e6, nu = 5)
mean(cos(very_long_chain))
```

```
## [1] 0.5258473
```

This gets close to the numerical value

```
integrate(function(x) cos(x)*dt(x, 5),-Inf,Inf)$value
```

```
## [1] 0.5239951
```