Basics of Quantifying Natural Hazards

MGEW23 WiSe 2017/18

Oliver Korup

January 23, 2018

Exponential Distribution

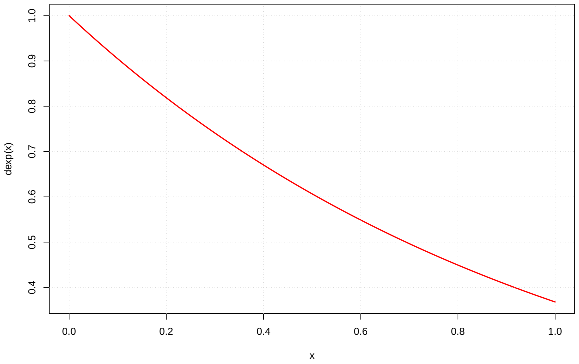
The **exponential distribution** derives directly from the exponential function $f(x) = e^x$, and has many important applications. The simplest form of the exponential distribution is:

$$p(x) = e^{-x}. \tag{1}$$

*Plot and verify with some **R** code that the equation above is indeed a probability density function.*

Exponential Distribution

```r
# Plot exponential distribution for x >= 0
curve(dexp(x), col = "red", lwd = 2); grid()
```



Exponential Distribution

```r
# Integrate to show that area beneath curve is 1
integrate(function(x) exp(-x), 0, Inf)
```

```
## 1 with absolute error < 5.7e-05
```

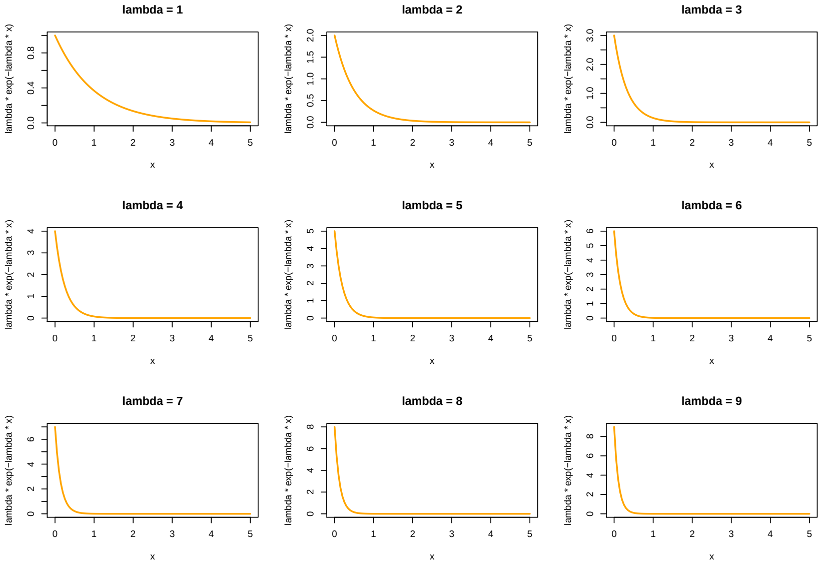
The general form of the exponential distribution is

$$p(x|\lambda) = \lambda e^{-\lambda x}, \tag{2}$$

where λ is the **rate**; the **mean** of the exponential distribution is $E(X) = \frac{1}{\lambda}$.

Plot the exponential distribution for different values of λ.
How does this change the distribution?

Exponential Distribution



Exponential Distribution

Note how λ controls the shape of the curve, and ensures normalisation: the area under the curve remains 1. We modify the exponential distribution further by shifting it by μ to include negative values:

$$p(x|\lambda, \mu) = \lambda e^{-\lambda(x-\mu)}, \tag{3}$$

Consider, for example:

```r
# Integrate to show that area beneath curve is 1
lambda <- 0.6; mu <- -12.2
integrate(function(x)
  lambda * exp(-lambda * (x - mu)), mu, Inf)
```

```
## 1 with absolute error < 1.3e-05
```

Exponential Distribution

How is the exponential distribution connected to the Poisson distribution? What are the implications?

Find and document at least five probability distributions that make use of, or are related to, the exponential distribution.

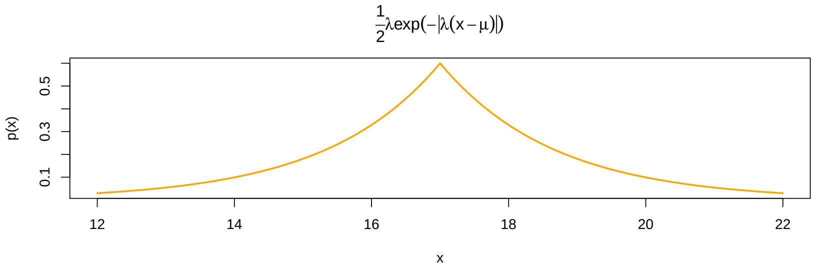
Exponential Distribution

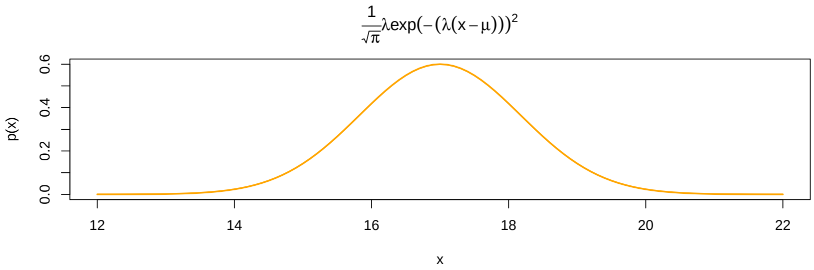
We use absolute values or squares on x to make the exponential distribution symmetric about μ, so that $-\infty < x < \infty$. In either case, we must normalise by 2 and $\sqrt{\pi}$, respectively, to ensure that the areas under the curves remain 1.

$$p(x|\lambda, \mu) = \frac{1}{2}\lambda e^{-|\lambda(x-\mu)|}, \tag{4}$$

$$p(x|\lambda, \mu) = \frac{1}{\sqrt{\pi}}\lambda e^{-[\lambda(x-\mu)]^2}. \tag{5}$$

Exponential Distribution





Exponential Family

Most of the probability distributions that we are dealing with are part of what is called the **exponential family**, and are built around an exponential function. Here we recall some useful maths:

$$e^a e^b = e^{a+b}. \tag{6}$$

and

$$\log(ab) = \log(a) + \log(b), \tag{7}$$

We will often be using the logarithmic function as the inverse of the exponential function, especially for dealing with very small probabilities and likelihoods.

Useful Maths

Now try this in **R**:

```r
log(27 * 4.2); log(27) + log(4.2)
```

```
## [1] 4.730921
```

```
## [1] 4.730921
```

```r
exp(-1.23) * exp(0.9); exp(-1.23 + 0.9)
```

```
## [1] 0.7189237
```

```
## [1] 0.7189237
```

Useful Maths

We can thus easily convert products to sums:

$$\prod_{i=1}^{n} \mathrm{e}^{a x_i} = \mathrm{e}^{\sum_{i=1}^{n} a x_i}, \tag{8}$$

and

$$\log \prod_{i=1}^{n} \mathrm{e}^{a x_i} = \sum_{i=1}^{n} a x_i. \tag{9}$$

Also note that

$$x^t = \mathrm{e}^{t \log(x)}. \tag{10}$$

Useful Maths

Here are some examples with the corresponding **R** code:

```
prod(exp(3.2 * 2:7)); exp(sum(3.2 * 2:7))
```

```
## [1] 3.334596e+37
```

```
## [1] 3.334596e+37
```

```
log(prod(exp(3.2 * 2:7))); sum(3.2 * 2:7)
```

```
## [1] 86.4
```

```
## [1] 86.4
```

Useful Maths

Keeping the following integrals handy will also be helpful:

$$\int_{-\infty}^{\infty} \mathrm{e}^{-x^2} \mathrm{d}x = \sqrt{\pi} \tag{11}$$

$$\int_{-\infty}^{\infty} \mathrm{e}^{-a(x+b)^2} \mathrm{d}x = \sqrt{\frac{\pi}{a}} \tag{12}$$

$$\int_{-\infty}^{\infty} \mathrm{e}^{-ax^2+bx+c} \mathrm{d}x = \sqrt{\frac{\pi}{a}} \mathrm{e}^{\frac{b^2}{4a}+c} \tag{13}$$

$$\int_{0}^{\infty} \mathrm{e}^{-ax^b} \mathrm{d}x = \frac{\Gamma(\frac{1}{b})}{ba^{\frac{1}{b}}} \tag{14}$$

Continuous Distributions

The **Weibull distribution** enjoys frequent use in extreme-value statistics, and comes in variants of two or three parameters:

$$P(x|\mu,\sigma,\alpha) = 1 - \exp\left[-\left[\frac{x-\mu}{\sigma}\right]^{\alpha}\right], \tag{15}$$

where $x \geq 0$, and $\alpha > 0$. The Weibull distribution simplifies to the exponential distribution for $\alpha = 1$.

When $\mu = 0$, and x measures the time to failure (e.g. repose intervals of volcanic eruptions), α holds information about the failure rate, while σ informs about the characteristic lifetime.

Continuous Distributions

The **Gamma distribution** can be expressed with different parameter sets. One way to write it is:

$$f_\gamma\left(x|a,b\right) = \frac{1}{b^a \Gamma\left(a\right)} x^{a-1} \mathrm{e}^{-\frac{x}{b}}, \tag{16}$$

where $x > 0$, and $a, b > 0$, and

$$\Gamma(\rho) = \int_0^\infty t^{a-1} \mathrm{e}^{-t}\, \mathrm{d}t, \tag{17}$$

where Γ is the Gamma function with $a > 0$.

The sum of a exponentially distributed variables is Gamma distributed, where a is a positive integer.

Continuous Distributions

The **Gaussian** or **Normal** distribution is one of the most important probability density functions (and a conjugate prior in Bayesian analysis):

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \qquad (18)$$

where μ is the **mean**, and σ is the **standard deviation**. Note that σ^2 is called *variance*, while σ^{-2} is called **precision**. For $\mu = 0$ and $\sigma = 1$ this probability density reduces to a Standard Normal distribution.

Gaussian Distribution

What is $p(x = 18.7|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

for a mean μ= 21.3 and a standard deviation σ= 4.9?

```r
x <- 18.7; mu <- 21.3; sigma <- 4.9
1 / (sqrt(2 * pi) * sigma) *
  exp(-0.5 * ((x - mu) / sigma) ^ 2)
```

```
## [1] 0.07072555
```

Now try the more convenient `dnorm()` command:

```r
dnorm(x, mu, sigma)
```

```
## [1] 0.07072555
```

Distributions in **R**

R has four commands for computing basic elements of probability distributions. For the Gaussian distribution, for example, we can

1. Compute the **probability density** $p(x)$:

```r
dnorm(13, mean = 20.1, sd = 4.5)
```

```
## [1] 0.02553495
```

2. Compute the **cumulative probability** $P(X \leq x)$:

```r
pnorm(13, mean = 20.1, sd = 4.5)
```

```
## [1] 0.05730834
```

Distributions in **R**

3. Compute the **quantile** of $q_{57\%}(x)$:

```r
qnorm(0.57, mean = 20.1, sd = 4.5)
```

```
## [1] 20.89368
```

4. Generate **random numbers** from the distribution:

```r
rnorm(4, mean = 20.1, sd = 4.5)
```

```
## [1] 26.95278 23.75168 27.84655 18.49942
```

Interpreting the Gaussian Distribution

The Gaussian distribution has many useful applications and interpretations. In particular, it can express

- the **natural variability** of a quantity (assuming the measurement errors are negligible);
- the **noise** (or inverse precision) of a measurement; and
- the distribution of **independent, additive contributions**.

Time for Some Exercise...

A group of geoscience students use a tape to measure the long axis of a coral boulder stranded on a sandy beach. The students have collected n data points of the axis diameter x. We know that the tape measure is only accurate to a certain point, and assume, for the sake of simplicity, that all measurements come from the same Gaussian distribution. We express the spread of measurements with the distribution's variance σ^2.

> *Given* n *measurements, what is the boulder's most believable actual size?*

Time for Some Exercise...

We use a brute-force approach to estimate the unknown mean μ using Bayes' Rule, and specify our initial beliefs about μ; note that σ^2 is known. We then generate *n* random numbers from the Gaussian distribution $\mathcal{N}\left(\mu, \sigma^2\right)$, apply Bayes' Rule, and check how accurately we can recover the true μ from this sample:

We define the true mean μ [m], which is the actual diameter of the coral boulder's long axis:

```r
# Define true mean 'mu' of Gaussian
mu <- 5.2
```

Time for Some Exercise...

We define the known, constant variance σ^2 [m^2], which expresses the accuracy of the tape measure:

```r
# Set known variance of Gaussian noise
sig <- 0.1
```

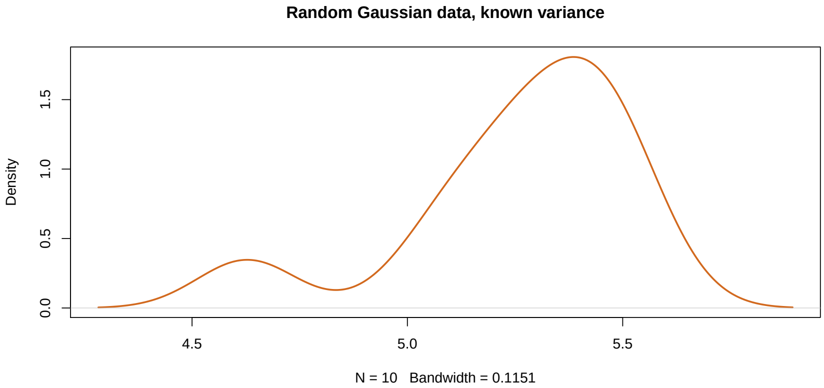
We then create *n* random data points from this distribution with `rnorm()`. Note that this command expects the *standard deviation* instead of the variance as an argument, so we need to take the square root of `sig`:

```r
# Create 'n' data points
n <- 10
dat <- rnorm(n, mu, sqrt(sig))
```

Time for Some Exercise...

Have a quick look at how these $n = 10$ data are distributed:

```r
# Plot a probability density estimate of the data
plot(density(dat), col = "chocolate", lwd = 2,
     main = "Random Gaussian data, known variance")
```



Time for Some Exercise...

To get started with using Bayes' Rule, we first specify our prior on μ, based on what we believe to be realistic for the maximum boulder diameter:

```r
# Set range of prior values
prior_mu <- seq(3.5, 6, 0.1)
```

Here, we selected 26 evenly spaced values between 3.5 m and 6 m as potential candidates for μ. We now assign weights to each of these values, reflecting our initial belief about each candidate value. In the simplest of cases, we could assign uniform weights to each of the 26 candidate means. This is also known as an **uninformed prior**.

Time for Some Exercise...

To demonstrate the power of Bayes' Rule, we choose instead random weights using the `runif()` command. Obviously this random assignment of weights hardly reflects any prior or expert knowledge and thus should be avoided otherwise:

```r
# Set weights for prior values
weights_mu <- runif(length(prior_mu))
```

Because the prior is a proper probability distribution, we need to make sure that all these weights add up to unity. The best way to guarantee this is to re-normalise so that all probability masses add up to unity:

```r
# Renormalise prior (to obtain a PDF)
p_mu <- weights_mu / sum(weights_mu)
```

Time for Some Exercise...

To compute the likelihood we use `dnorm()`, assuming all measurements are independent and from an identical distribution (**i.i.d.**). Thus we compute the likelihood of observing those data by multiplying their individual probability densities. With the `prod()` command we obtain the overall likelihood for the 26 candidate means:

```r
# Generate empty container for likelihood
p_data_mu <- rep(0, length(prior_mu))

# Loop through each candidate mean in the prior
i <- 1
while (i <= length(prior_mu)) {
  p_data_mu[i] <- prod(dnorm(dat,
                             mean = prior_mu[i],
                             sd = sqrt(sig)))
  i <- i + 1
}
```

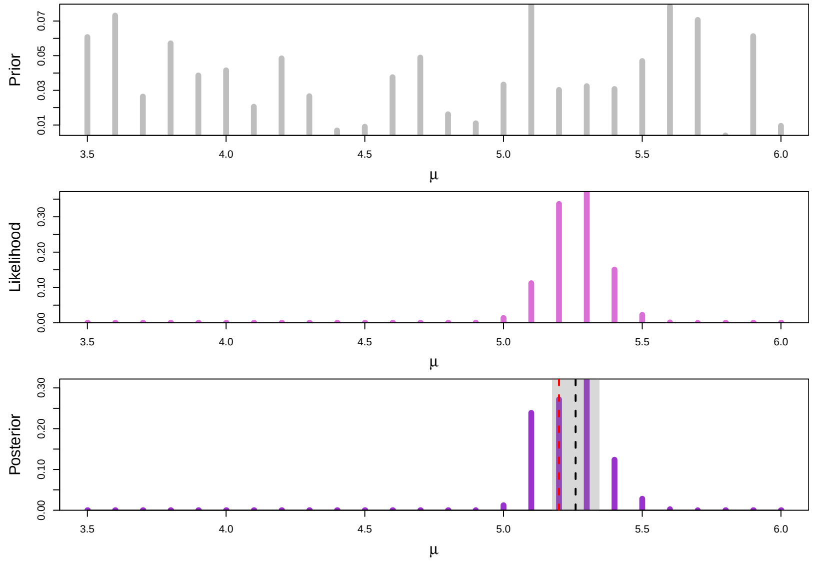
Time for Some Exercise...

Finally, we apply Bayes' Rule and compute the posterior. Note that we do not compute the **evidence** $p(\mathcal{D})$ explicitly; instead, we simply re-normalise the posterior:

```r
# Compute and renormalise posterior
p_mu_data <- p_data_mu * p_mu
p_mu_data <- p_mu_data / sum(p_mu_data)
```

Now plot the results. Use a plot containing three row-wise panels. Also add to the posterior distribution the ***frequentist sample mean and standard error*** *as a vertical line and a grey box, respectively; a red vertical line should show the true mean.*

Time for Some Exercise...



Maximum Likelihood Estimator

The likelihood for observing data $\mathcal{D}$ from a **homoscedastic** Gaussian distribution (variance σ^2 is known and constant) is:

$$\mathcal{L} = p(\mathcal{D}|\mu,\sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2}. \tag{19}$$

To make things computationally feasible, we set the derivative of the **log-likelihood** to zero, and find which μ maximises $\mathcal{L}$:

$$\frac{d \ln \mathcal{L}}{d\mu} \equiv 0 \tag{20}$$

.

Maximum Likelihood Estimator

$$\ln \mathcal{L} = \ln \frac{1}{\left(\sqrt{2\pi}\sigma\right)^n} - \sum_{i=1}^{n} \frac{1}{2} \left(\frac{x_i - \mu}{\sigma} \right)^2 . \qquad (21)$$

For $\frac{d \ln \mathcal{L}}{d\mu} \equiv 0$ we obtain:

$$0 = \sum_{i=1}^{n} \frac{x_i - \mu_0}{\sigma^2}, \text{and thus} \sum_{i=1}^{n} \mu_0 = n\mu_0 = \sum_{i=1}^{n} x_i, \qquad (22)$$

We see that $\mu_0 = \frac{1}{n} \sum_{i=1}^{n} x_i$, or the **arithmetic mean** of the data, is the maximum likelihood for a Gaussian with fixed variance.

Maximum Likelihood Estimator

For the MLE of the mean of a **heteroscedastic** Gaussian distribution (variance σ_i^2 is known and variable) we obtain with $\frac{d \ln \mathcal{L}}{d\mu} \equiv 0$:

$$0 = \sum_{i=1}^{n} \frac{x_i - \mu_0}{\sigma_i^2}, \tag{23}$$

and thus

$$\mu_0 = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}, \tag{24}$$

where $w_i = \sigma_i^{-2}$, so that the MLE of the mean is weighted by the different precisions.

Time for Some Exercise...

Let us assume that we record annual maximum storm-wave heights, and we wish to learn the period θ of wave peaks from the data. We also have some previous knowledge about θ, which we express in a prior distribution, and assume i.i.d. data. We first generate an artificial time series from parameters that we define beforehand. That way we can test how reliably the Bayesian approach captures the true values despite some noise in the measurements and despite some possibly vague prior beliefs about θ.

www.nationalgeographic.com

Time for Some Exercise...

We start off by defining the real wave period θ:

```
# Define real wave period in some unit time [T]
theta <- 0.3
```

Then we specify the period of observation t:

```
# Define time interval
t <- 0:99
```

Time for Some Exercise...

Let us assume that we know the accuracy of our instrumental measurements, and that we can describe this with a Gaussian distribution: we assume that our measurements have a constant variance σ^2 [m^2]:

```r
# Define variance of measurements (Gaussian noise)
sigma <- 5
noise <- rnorm(length(t), 0, sqrt(sigma))
```

Now we can create the periodic, though noisy, wave data. We use a periodic function such as `sin()` for this purpose:

```r
# Create data with Gaussian noise
x <- sin(theta * t) + noise
```

Note that the noise applies to the measurements of wave height [m] about some arbitrary mean by definition.

Time for Some Exercise...

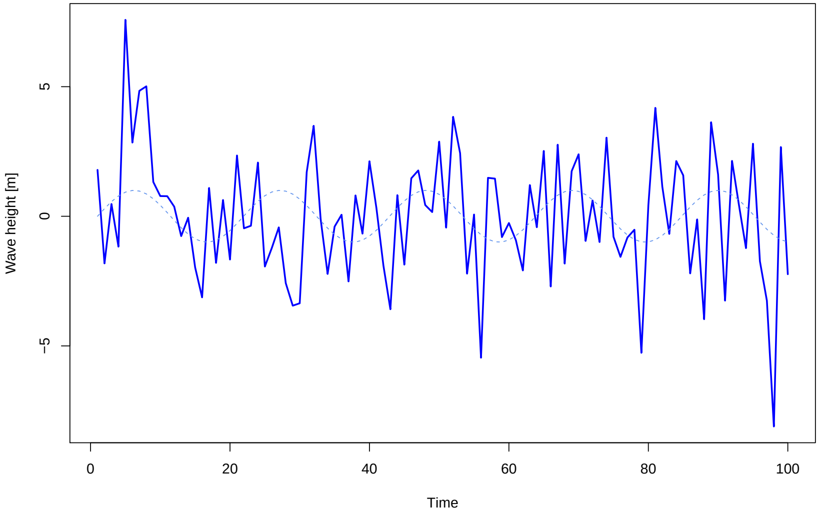
We plot the observed data with noise, and also add a dashed line showing the data without any noise.

```r
# Plot time series
plot.ts(x, col = "blue", lwd = 2,
        xlab = "Time",
        ylab = "Wave height [m]",
        cex.lab = 1.5)

# Add clean data without measurement noise
lines(sin(theta * t), lty = 2, col = "cornflowerblue")
```

Note that we use the `plot.ts()` command for plotting the time series. This command expects a single vector of measurements that we assumed were taken at regular intervals.

Time for Some Exercise...



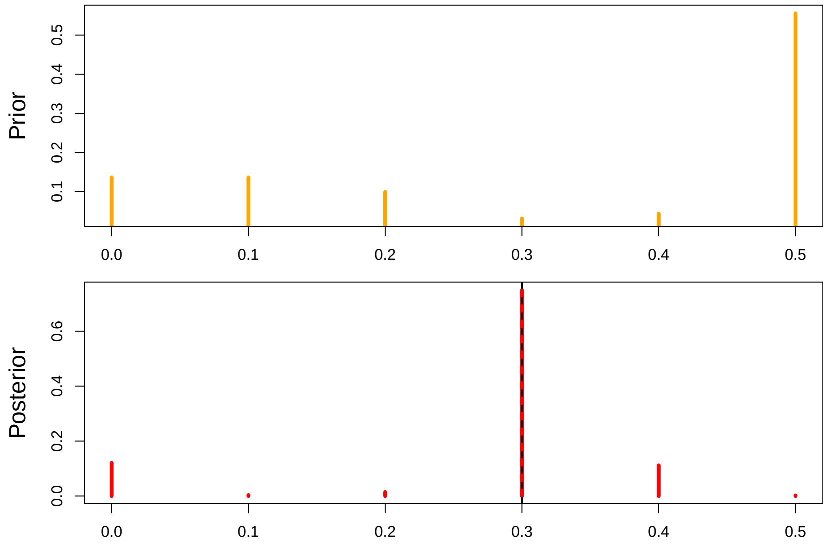
Time for Some Exercise...

We now define and normalise our prior belief about the wave-peak period θ:

For each of these values of θ we compute the likelihood given the observed data. We explicitly encode the **likelihood function**, which varies with the input of θ. Here we use the `function()` command to specify our own **R** command:

```r
# Define likelihood p(data|theta) function
p_data_given_theta <- function(theta){
  prod(exp(-(x - sin(theta * t)) ^ 2 /
              (2 * sigma)) / sqrt(2 * pi * sigma))
}
```

Time for Some Exercise...


Prior
Posterior