

CTE Constant Temperature Equivalent Vignette

Donald T. McKnight

2026-01-19

Contents

Background and formulas	1
Example	2

Background and formulas

This package implements the formulas for the Constant Temperature Equivalent (CTE) for egg incubation derived by Georges (1989). Please see that paper for additional information on the theory and math behind CTE.

In short, CTE takes data on the fluctuations in natural nests and translates the data into a single constant value (the CTE) that would produce the same rate of egg development. Stated another way, CTE is the temperature above which half of embryonic development occurs. Thus, eggs in an incubator set to a given CTE will develop at the same rate as the eggs in the natural nest from which the CTE was calculated.

Three input values are needed to calculate CTE for a given day:

- M = mean temperature
- R = the maximum deviation of temperatures from M (e.g., if the range of temperatures is 14–16 and $M = 20$, then $R = 6$)
- T_0 = the temperature below which no development occurs (this value should be based on a general knowledge of the species being studied)

Using those inputs, CTE can be calculated by first describing the daily sinusoidal variation in nest temperature using Formula 1, where, for a given day, T is the temperature (in degrees centigrade) at a given time of day (position along the curve), and t is the time of day (position along the curve) expressed as a range from 0– 2π (e.g., 06:00 = $\pi/2 = 1.57$, 12:00 = π , and 18:00 = $1.5\pi = 4.71$).

$$\text{Formula1 : } T = R\cos(t) + M$$

If we know the point along the curve above which half of development occurs (t'), then the corresponding temperature at that time (T') will be the CTE (Formula 2).

$$\text{Formula2 : } \text{CTE} = R\cos(t') + M$$

We can use Formula 3 to find t' , except for cases where the minimum daily temperature (as calculated by $M-R$) is lower than T_0 , in which case Formula 4 is needed, where t_0 is the time (point along the curve) when the temperature for a given day first drops below T_0 .

$$Formula3 : t' = \frac{pi}{2} - \frac{R}{(M - T_0)} \sin(t')$$

$$Formula4 : t' = \frac{t_0}{2} + \frac{R}{2(M - T_0)} \sin(t_0) - \frac{R}{(M - T_0)} \sin(t')$$

Formulas 3 and 4 have to be solved iteratively. The CTE function automatically detects which formula to use based on whether temperatures are ever less than T_0 (i.e., $M-R < T_0$). Formula 3 can easily be solved iteratively by using a starting seed for t' (any value > 0 and $< 2\pi$) on the right side of the equation to calculate a new t' (left side of the equation) then using the new t' in the formula (right side) for the next iteration and repeating the process until t' is equal on both sides of the equation (the CTE function defines “equal” as equal to five decimal places).

For Formula 4, a strictly iterative approach (as described above) sometimes fails to find a stable solution. Therefore, as recommended in Georges et al. (2004), the CTE function implements a binary search algorithm to achieve a stable result.

Suggested reading:

- Georges A (1989) Female turtles from hot nests: is it duration of incubation or proportion of development at high temperatures that matters? *Oecologia* 81:323–328.
- Georges A, Doody S, Beggs K, Young J (2004) Thermal models of TSD under laboratory and field conditions. In: Temperature Dependent Sex Determination. Valenzuela N, Lance VA (eds) Smithsonian Books, Washington DC, p 79–91
- Georges A, Limpus C, Stoutjesdijk R (1994) Hatchling sex in the marine turtle *C. caretta* is determined by proportion of development at a temperature, not daily duration of exposure. *J Exp Zool* 270:432–444.

Example

The example.temp.data object contains a hypothetical example of two nests over 1 or 2 days, with temperature data recorded every half hour. Let’s first take a single nest for a single date and look at how we would run the function manually.

```
library("CTE")
library("dplyr")
#>
#> Attaching package: 'dplyr'
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
data(example_temp_data)

#extract data for nest 2, which only has 1 day of data
nest2 <- example_temp_data[example_temp_data$nest == "nest2",]

#calculate the mean (M)
daily.mean <- mean(nest2$temp)

#calculate the maximum deviation from the mean (R)
```

```

#take the difference between the maximum and mean temp and difference between mean
# and minimum temp. The larger of the two values will be R.
daily.dev <- max(
  (max(nest2$temp)-daily.mean),
  (daily.mean-min(nest2$temp))
)

#The T0 value (temp below which no development occurs) cannot be calculated from the
# data and has to be determined based on the biology of the study species.
# We will use T0 = 16 for our example.
#We will run CTE using the calculated values and leave t and max.it on their defaults
cte.res <- CTE(M = daily.mean, R = daily.dev,T0 = 16)

cte.res
#> [1] 26.26212

```

In this example, 26.26 is the CTE for that day. In other words, if an incubator had been set to a constant temperature of 26.26, the same amount of development would have occurred in eggs in the incubator as eggs in this wild nest.

Note that because the calculations are iterative, we would get the exact same result if we specified a t other than the default, but the run time will change slightly.

```

cte.res2 <- CTE(M = daily.mean, R = daily.dev,T0 = 16,t=.01)

cte.res
#> [1] 26.26212
cte.res2
#> [1] 26.26212

```

In many cases, you will want to run this automatically over multiple nests or days. I'll demonstrate one way to do this using the dplyr package.

```

#calculate M (daily mean) and R (daily.dev) for each nest for each date.
daily.values <- example_temp_data %>%
  group_by(nest,date) %>%
  dplyr::summarize(daily.mean = mean(temp),
                  daily.dev = max(
                    (max(temp)-mean(temp)),
                    (mean(temp)-min(temp))
                  ))
#> `summarise()` has grouped output by 'nest'. You can override using the
#> `.groups` argument.

daily.values
#> # A tibble: 3 x 4
#> # Groups:   nest [2]
#>   nest    date    daily.mean  daily.dev
#>   <chr> <chr>     <dbl>      <dbl>
#> 1 nest1 2021-11-05    20.1      6.30
#> 2 nest1 2021-11-06    20.7      4.03
#> 3 nest2 2021-11-06    21.5      6.60

```

```

#calculate CTE for each nest for each date
#CTE will be added as a new column
daily.values <- daily.values %>%
  group_by(nest,date) %>%
  dplyr::mutate(CTE=CTE(M = daily.mean,
                        R=daily.dev,T0=16))

daily.values
#> # A tibble: 3 x 5
#> # Groups: nest, date [3]
#>   nest    date   daily.mean daily.dev     CTE
#>   <chr> <chr>     <dbl>      <dbl>     <dbl>
#> 1 nest1 2021-11-05    20.1      6.30    24.9
#> 2 nest1 2021-11-06    20.7      4.03    23.2
#> 3 nest2 2021-11-06    21.5      6.60    26.3

```

This returns the CTE for each nest for each date