

Threshold models are very common in biology, where they can be successfully used to describe the daily (or hourly) progress towards a certain developmental stage, as influenced by the environmental conditions, mainly humidity and temperature. In this post I will show examples relating to seed germination, but the very same concepts also apply to the growth of plants and other organisms. In biology, these models are often known as thermal-time or hydro-time models.

Considering seed germination, the response variable is the Germination Rate (GR), that is the reciprocal of germination time; for example, if a seed takes 7 days to accomplish the germination process, the GR is equal to $(1/7 = 0.143)$ and it represents the fraction of germination that is accomplished in one day. The GR is a good measure of velocity: the higher the value the higher the speed. If we plot the GR against, e.g., temperature, we should very likely observe a response pattern as in Figure 1; the three thresholds are, respectively, the *base temperature* ($T_{b_}$), the *optimal temperature* ($T_{o_}$) and the *ceiling temperature* ($T_{c_}$). If we look at the effect of soil humidity on GR, we should expect a response pattern with only one threshold, i.e. the *base water potential* (e.g. the first half of Figure 1, up to the maximum response level).

Although the equations I am going to introduce are commonly used in the seed science literature, I am confident that you might find them useful for a lot of other applications. For all those equations, I have built the related R functions, together with self-starting routines, which can be used for nonlinear regression analyses with the `drm()` function in the 'drc' package (Ritz et al., 2019). The availability of self-starting routines will free you from the hassle of having to provide initial guess for model parameters. All these R functions are available within the 'drcSeedGerm' package (Onofri, 2019).

The post is rather long, but you do not need to read it all; have a look at the graph below to spot the shape you are interested in and use the link to reach the relevant part in this web page. But, first of all, do not forget to install (if necessary) and load the 'drcSeedGerm' package, by using the code below.

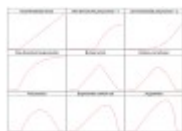
```
#installing package, if not yet available
# library(devtools)
# install_github("onofriandreapg/drcSeedGerm")

# loading package
library(drcSeedGerm)
```

List of threshold models

In this post I will show the following threshold models:

1. [One-threshold linear](#)
2. [One-threshold polynomial – 1](#)
3. [One-threshold polynomial – 2](#)
4. [One-threshold exponential](#)
5. [Broken-stick model](#)
6. [Broken-curvilinear model](#)
7. [Polynomial model](#)
8. [Exponential switch-off model](#)
9. [Hyperbolic model](#)



One-threshold linear model

In some cases, we may need to model a process occurring only above a certain threshold level for the predictor variable. Models of this kind have been used to describe the effect of environmental humidity (Ψ , in MPa) on germination rate (Bradford, 2002):

$$GR = \frac{\max[\Psi, \Psi_b]}{\theta_H} \quad \quad \quad (1)$$

The parameter Ψ_b is the *base water potential* (in MPa), representing the minimum level of humidity in the substrate to trigger the germination process. The other parameter θ_H (in MPa day or MPa hour) is the hydro-time constant.

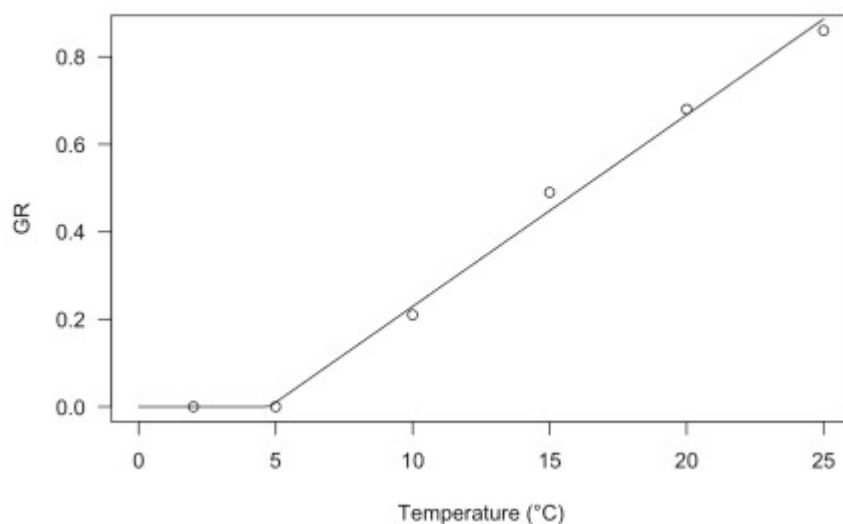
A totally similar model has been used by Garcia-Huidobro et al (1982), to describe the effect of sub-optimal temperatures (T in $^{\circ}\text{C}$) on the germination rate:

$$GR = \frac{\max[T, T_b]}{\theta_T} \quad \quad \quad (2)$$

where T_b is the base temperature and θ_T is the thermal time (in $^{\circ}\text{C d}$).

Both models can be fitted in R, by using the two functions `GRPsi.Lin()` and `GRT.GH()`; they are totally equivalent, apart from parameter names. In the example below I have fitted the second equation to a seed germination dataset; this type of data is usually heteroscedastic, thus, please note the use of a robust variance-covariance estimator for model parameters (Zeileis, 2006).

```
Tlev <- c(2, 5, 10, 15, 20, 25)
GR <- c(0, 0, 0.21, 0.49, 0.68, 0.86)
modGH <- drm(GR ~ Tlev, fct = GRT.GH())
library(sandwich); library(lmtest)
coeftest(modGH, vcov = sandwich)
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## Tb:(Intercept)    4.77160    0.31993  14.915 0.0001177 ***
## ThetaT:(Intercept) 22.83118    0.62847  36.328 3.428e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
plot(modGH, log="", xlim = c(0, 25), legendPos = c(5, 1.2),
     xlab = "Temperature ( $^{\circ}\text{C}$ )")
```



One threshold polynomial

In my experience, I have found that the relationship between GR and water potential in the substrate may, sometimes, be curvilinear. For these situations, I have successfully used the following equations:

$$[GR = \frac{\max\left[\Psi, \Psi_b\right]^2 - \Psi_b^2}{\theta_H} \quad \quad \quad (3)]$$

and:

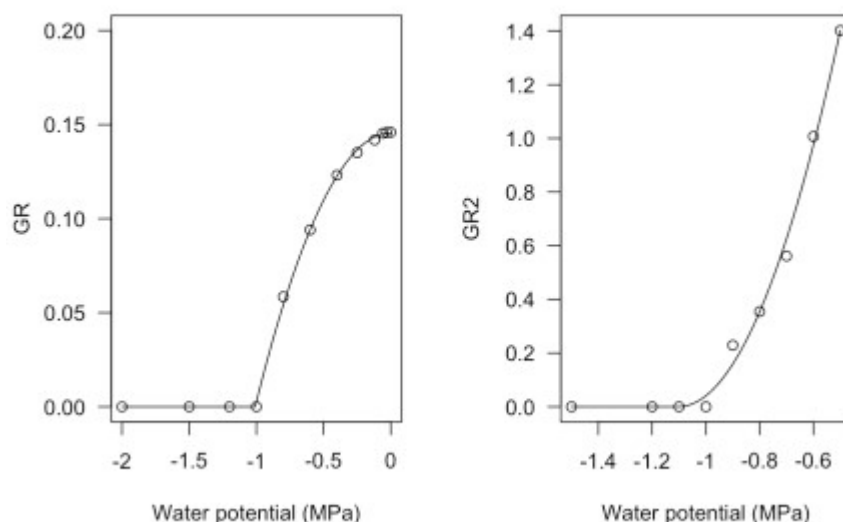
$$[GR = \frac{\left(\max\left[\Psi, \Psi_b\right] - \Psi_b\right)^2}{\theta_H} \quad \quad \quad (4)]$$

Both models can be fitted in R, by using the two functions `GRPsi.Pol()` and `GRPsi.Pol2()`, as shown in the box below.

```
# Observed data
Psi <- c(-2, -1.5, -1.2, -1, -0.8, -0.6, -0.4, -0.25,
        -0.12, -0.06, -0.03, 0)
GR <- c(0, 0, 0, 0, 0.0585, 0.094, 0.1231, 0.1351,
        0.1418, 0.1453, 0.1458, 0.1459)
Psi2 <- c(-0.5, -0.6, -0.7, -0.8, -0.9, -1, -1.1, -1.2,
        -1.5)
GR2 <- c(1.4018, 1.0071, 0.5614, 0.3546, 0.2293, 0, 0,
        0, 0)

modHT <- drm(GR ~ Psi, fct = GRPsiPol())
modHT2 <- drm(GR2 ~ Psi2, fct = GRPsiPol2())
par(mfrow = c(1,2))

plot(modHT, log="", legendPos = c(-1.5, 0.15),
     ylim = c(0, 0.20), xlab = "Water potential (MPa)")
plot(modHT2, log="", legendPos=c(-1.3, 1),
     xlab = "Water potential (MPa)")
```



One threshold: exponential

All the previous models tend to go up to infinity when the predictor value (temperature or water potential) goes to infinity. In some instances, we may need an asymptotic model, for example, to describe the oxygen uptake kinetics during a walk test (see Baty et al., 2014, although their threshold model is more complex) or to describe the response of the maximum proportion of germinated seeds to soil humidity (Onofri et al., 2018).

In practice, we could use a shifted exponential model:

$$\pi = G \left[1 - \exp \left(- \frac{\max[\Psi, \Psi_b]}{\sigma} \right) \right] \quad (5)$$

where π is the proportion of germinated seeds, G is the fraction of non-germinable seeds (e.g., dormant seeds) and σ describes how quickly the population of seeds responds to increased humidity in the substrate.

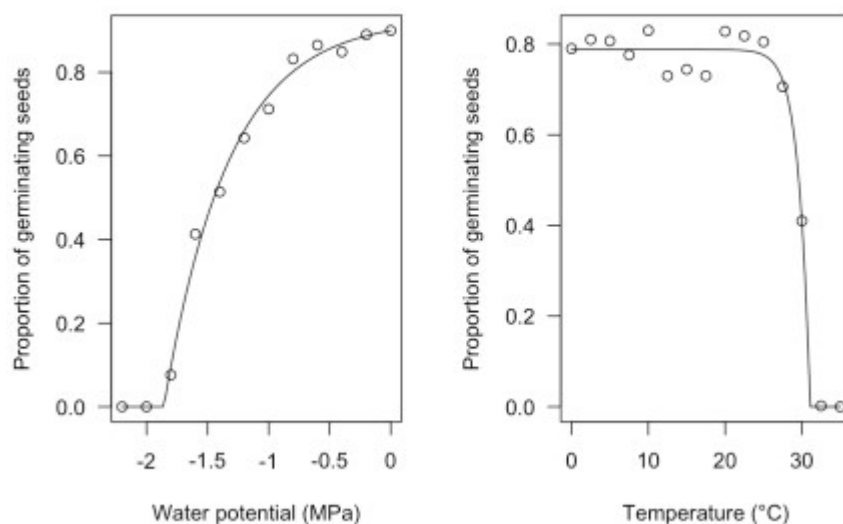
If we reverse the sign of σ in Eq. 5, we obtain a decreasing trend, which might be useful to describe the effect of super-optimal temperatures on the proportion of germinated seeds, going down to 0 at the ceiling temperature threshold.

Both equations can be fitted using the self-starters `PmaxPsi1()` and `PmaxT1()`. The two R functions are totally equal, apart from parameters names.

```
par(mfrow = c(1,2))
# Pmax vs Psi
Psi <- seq(-2.2, 0, by = 0.2)
Pmax <- c(0, 0, 0.076, 0.413, 0.514, 0.643, 0.712,
          0.832, 0.865, 0.849, 0.89, 0.90)
mod <- drm(Pmax ~ Psi, fct = PmaxPsi1())
plot(mod, log = "", xlab = "Water potential (MPa)",
      ylab = "Proportion of germinating seeds")

# Pmax vs temperature
Tval <- c(0, 2.5, 5, 7.5, 10, 12.5, 15, 17.5,
          20, 22.5, 25, 27.5, 30, 32.5, 35)
```

```
Pmax2 <- c(0.79, 0.81, 0.807, 0.776, 0.83,
           0.73, 0.744, 0.73, 0.828, 0.818,
           0.805, 0.706, 0.41, 0.002, 0)
mod2 <- drm(Pmax2 ~ Tval, fct = PmaxT1())
plot(mod2, log = "", xlab = "Temperature (°C)",
      ylab = "Proportion of germinating seeds")
```



‘Broken-stick’ model

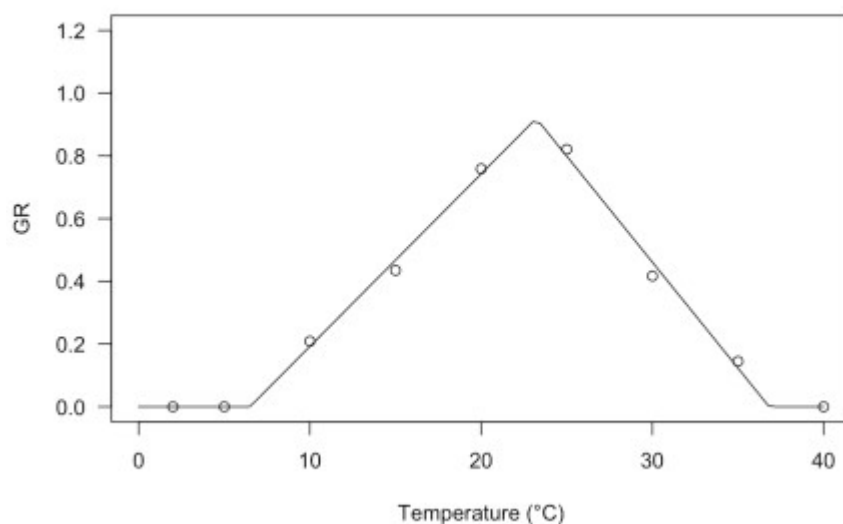
A broken-stick trend, as the one depicted in the Figure 1 was used by Alvarado and Bradford (2002) to model the effect of temperature on the germination rate. The equation is:

$$\left[GR = \frac{\min \left[T, T_o \right] - T_b}{\theta_T} \right], \left\{ 1 - \frac{\min \left[\max \left[T, T_o \right], T_c \right] - T_o}{T_c - T_o} \right\} \quad \quad (6)$$

Base, optimal and ceiling temperatures (respectively (T_b) , (T_o) and (T_c)) are included as parameters, together with the hydro-thermal time parameter (θ_T) . Equation 6 can be easily fitted with the [GRT.BS\(\)](#) function in the ‘drcSeedGerm’ package.

```
Tval <- c(2, 5, 10, 15, 20, 25, 30, 35, 40)
GR <- c(0, 0, 0.209, 0.435, 0.759, 0.821, 0.417, 0.145, 0)

modBS <- drm(GR ~ Tval, fct = GRT.BS\(\))
plot(modBS, log="", xlim = c(0, 40), ylim=c(0,1.2),
      legendPos = c(5, 1.0), xlab = "Temperature (°C)")
```



```
coeftest(modBS, vcov = sandwich)
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## Tc:(Intercept)   36.81971    0.41113   89.556 3.290e-09 ***
## Tb:(Intercept)    6.49657    0.43146   15.057 2.340e-05 ***
## To:(Intercept)   23.21649    0.29481   78.750 6.256e-09 ***
## ThetaT:(Intercept) 18.18250    0.76345   23.816 2.431e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Broken-curvilinear model

Broken-stick trends may not be reasonable for biological processes, which might be better described by curvilinear equations. Rowse and Finch-Savage (2003) proposed another equation with two components: the first one depicts a linear increase of the GR value with temperature, which is off-set by the second component, starting from $(T = T_d)$, which is close to (but not coincident with) (T_o) . The equation is:

$$GR = \frac{\max\left[T, T_b\right] - T_b}{\theta_T} \left\{ 1 - \frac{\min\left[\max\left[T, T_d\right], T_o\right] - T_d}{T_c - T_d} \right\} \quad (7)$$

The optimal temperature can be derived as:

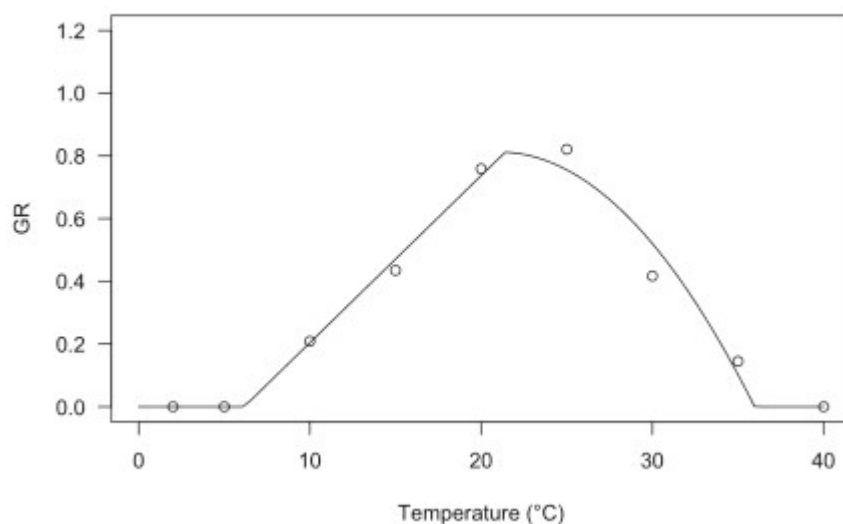
$$T_o = \frac{1 + kT_b + kT_d}{2k}$$

where:

$$k = \frac{1}{T_c - T_d}$$

For this equation, you will find the `GRT.RFb()` self-starter in the 'drcSeedGerm' package.

```
modRF <- drm(GR ~ Tval, fct = GRT.RFb())
plot(modRF, log="", xlim = c(0, 40), ylim=c(0,1.2),
      legendPos = c(5, 1.0), xlab = "Temperature (°C)")
```



Polynomial model

According to Mesgaran et al (2017), the negative and positive effects of temperature coexist for all temperatures above (T_b) . Their proposed equation can be parameterised as:

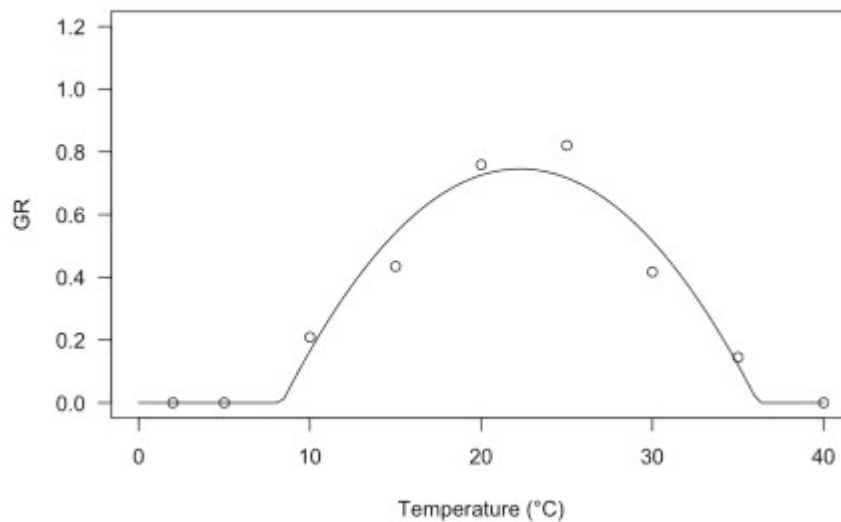
$$GR = \frac{\max \left[T, T_b \right] - T_b}{\theta_T} \left[1 - \frac{\min \left[T, T_c \right] - T_b}{T_c - T_b} \right]^2 \quad (8)$$

It is easy to see that the GR value is a second-order polynomial function of $(T - T_b)$ and, therefore, the curve is symmetric around the optimal temperature value, which can be derived as:

$$T_o = \frac{T_c - T_b}{2} + T_b$$

For this equation, the self-starting function is `GRT.M()`.

```
modM <- drm(GR ~ Tval, fct = GRT.M())
plot(modM, log="", xlim = c(0, 40), ylim=c(0,1.2),
      legendPos = c(5, 1.0), xlab = "Temperature (°C)")
```



Exponential switch-off model

Equations 6 to 8 use a product, wherein the first term represents the accumulation of thermal time and the second term may be seen as a switch-off term that is 1 either when $(T < T_o)$ (Equation 6) or $(T < T_d)$ (Equation 7) or $(T = T_b)$ (Equation 8) and decreases progressively as temperature increases. In all the above equations, the switch-off term is linear, although we can use other types of switch-off terms, to obtain more flexible models. One possibility is to use an exponential switch-off term, as in the equation below:

$$GR = \frac{\max[T, T_b] - T_b}{\theta_T} \left\{ \frac{1 - \exp\left[-k(\min[T, T_c] - T_c)\right]}{1 - \exp\left[-k(T_b - T_c)\right]} \right\} \quad (9)$$

where k is the switch-off parameter: the lower the value, the higher the negative effect of temperature at super-optimal levels. The response of GR to temperature is highly asymmetric with a slow increase below (T_o) and a steep drop afterwards.

I have successfully used this model in Catara et al (2016) and Masin et al (2017). The self-starting function in R is `GRT.Exb()`.

```
Tval <- c(2, 5, 10, 15, 20, 25, 30, 35, 40)
```

```
GR <- c(0, 0, 0.209, 0.435, 0.759, 0.821, 0.917, 0.445, 0)
```

```
modExb <- drm(GR ~ Tval, fct = GRT.Exb())
```

```
summary(modExb)
```

```
##
```

```
## Model fitted: Exponential effect of temperature on GR50 (Type II -  
Masin et al., 2017)
```

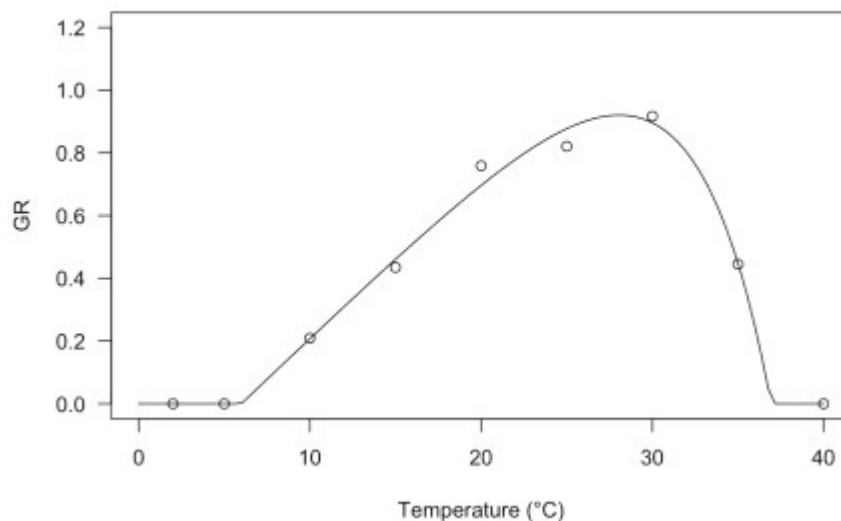
```
##
```

```
## Parameter estimates:
```

```
##
```

	Estimate	Std. Error	t-value	p-value	
Tb:(Intercept)	5.991778	1.205747	4.9693	0.0042141	**
ThetaT:(Intercept)	19.254345	2.795867	6.8867	0.0009881	***
k:(Intercept)	0.181612	0.060564	2.9987	0.0301453	*
Tc:(Intercept)	36.933370	0.501666	73.6214	8.758e-09	***


```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error:
##
## 0.04049231 (5 degrees of freedom)
plot(modExb, log="", xlim = c(0, 40), ylim=c(0,1.2),
     legendPos = c(5, 1.0), xlab = "Temperature (°C)")
```



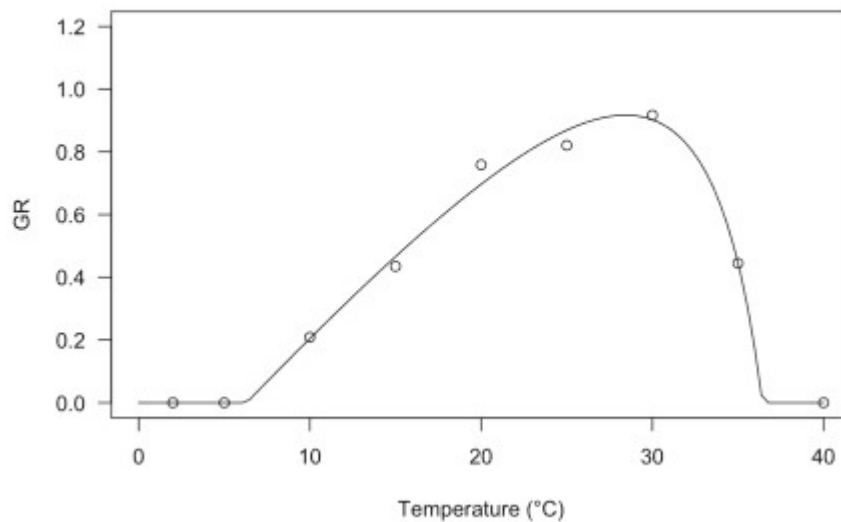
Hyperbolic model

The following model was derived by the simple yield loss function devised by Kropff and van Laar (1993). It is very flexible, as it may depict different types of relationships between temperature and base water potential, according to the value taken by the parameter $\backslash(q\backslash$.

$$\backslash[GR(g, T) = \frac{\backslashmax \backslashleft[T, T_b\right] - T_b\}{\theta_T} \backslashleft(1 - \frac{q}{\frac{\backslashmin \backslashleft[T, T_c\right] - T_b\}{T_c - T_b}} \right) \backslashquad \backslashquad \backslashquad (10)\backslash]$$

In R, this model can be fitted by using the self-starter `GRT.YL()` and we see that, with our dataset, the fit is very much like that of the exponential switch-off function.

```
modYL <- drm(GR ~ Tval, fct = GRT.YL())
plot(modYL, log="", xlim = c(0, 40), ylim=c(0,1.2),
     legendPos = c(5, 1.0), xlab = "Temperature (°C)")
```



A warning message

When we collect data about the response of germination rates to temperature and use them to parameterise nonlinear regression models by using nonlinear least squares, the basic assumption of homoscedasticity is rarely tenable. **We should not forget this!** in the above example I used a robust variance-covariance sandwich estimator, although other techniques can be successfully used to deal with this problem.