Dear Emma,

This was a challenging task. To really understand the R and statistical approach I needed to dig into things a bit, and I'm concerned about what I found. With all due respect to your colleagues, I think there are some very questionable steps in the outlined approach. I'm not the sharpest tool in the shed, so maybe if you can address my ignorance it will cover a lot of other people as well. Here are some issues.

General Comments

1) The data do not appear to have come from the cited paper, which only ran trials for 25 days, not 43, on only one species. In addition, in Fernandez-Pascual et al. (2015) no seeds germinated at a constant temperature, whereas that's the only experiment you conducted. This makes it difficult to understand the time-stamp associated with the example data. What exactly does time = 0 mean? One likely possibility is elapsed time since the experiment began; G = 0 at time 0 would mean that no seeds were already germinated when placed in the petri dish, and that the experiment ran for 43 days. If this interpretation is correct, then the data as analyzed are incorrect. Data.frame `dat` (the raw data) has times from 0 – 43. This would seem to imply elapsed number of days since the start of the experiment. However, data.frame `dat1` says that the first observation ran from time 0 to time 1, which is not correct. `dat1` then goes on to say that the final observation is 100% germination at time ∞. This is clearly not the case and cannot be correct. Even given infinite time not all of the seeds will germinate and you cannot know the right answer even if you dissect all the ungerminated seeds at day 43. This all seems contrived to use an "event" approach, but the time intervals are not calculated correctly. This causes issues later on in fitting the dose/response models as you will see.

2) Next, to make sure you have spanned an appropriate range of temperatures in your experiment you summarize the confidence intervals of a binomial for the final germination counts for each experiment (data.frame `FGP`). This would be much easier to visualize if `FGP` was separated by species and ordered by Treatment.

```
FGP[FGP$Grouping=='A',]
   Grouping Treatment method  x  n mean     lower     upper
1         A     11.25  logit  0 25 0.00 0.0000000 0.1371852
2         A     13.75  logit  0 25 0.00 0.0000000 0.1371852
3         A     16.25  logit  0 25 0.00 0.0000000 0.1371852
4         A     18.75  logit  7 25 0.28 0.1397342 0.4821515
5         A     21.25  logit  7 25 0.28 0.1397342 0.4821515
6         A     23.75  logit 22 25 0.88 0.6870053 0.9607855
7         A     26.25  logit 22 25 0.88 0.6870053 0.9607855
8         A     28.75  logit 25 25 1.00 0.8628148 1.0000000
9         A     31.25  logit 19 20 0.95 0.7177966 0.9930035
10        A     33.75  logit  9 25 0.36 0.1990861 0.5600309
```

What you can then easily see is that for species `A` no germination took place below 18.75°C, but that species `A` was still germinating at the maximum experimental temperature of 33.75°C, making it impossible to know T_c. Your derived estimate of T_c is an extrapolation of a linear model fitted to a truncated series of very few points. I think it's quite problematic.

```
FGP[FGP$Grouping=='B',]
   Grouping Treatment method  x  n mean      lower     upper
11        B     13.75  logit  0 25 0.00 0.00000000 0.1371852
12        B     16.25  logit  2 25 0.08 0.02008974 0.2694446
13        B     18.75  logit 17 25 0.68 0.47837454 0.8311921
14        B     21.25  logit 24 25 0.96 0.76452932 0.9943948
15        B     23.75  logit 25 25 1.00 0.86281483 1.0000000
16        B     26.25  logit  0 25 0.00 0.00000000 0.1371852
17        B     28.75  logit  0 25 0.00 0.00000000 0.1371852
18        B     31.25  logit  0 25 0.00 0.00000000 0.1371852
```

For species `B` you have both T_b and T_c. This result is the opposite of what you state on page 7 lines 4–17. It's also easily seen in Fig. 1. It's true that there are no intermediate results between T_o and T_c for species `B`, but I don't see how that invalidates the estimate as stated in the manuscript. It seems preferable to species A in my view.

3) I cannot understand how it is legitimate to fit a dose-response model to these data. The number of germinations at time T_x is not independent of the number of germinations at time $T_{!x}$. Because seeds cannot germinate twice, the number of possible germinations declines over time even though the "dose" is effective and positive. I'm certainly not an expert in dose-response experiments, but isn't the assumption is that the dose/response couplets are independent of all other does/response couplets? This seems like a time-series problem, not a dose-response problem. I realize that Ritz et al. (2015) mention germination trials as an application, but in that case I would think you want one dish run for one day, a different dish run for two days, etc. up to 43 dishes for each temperature so that you have some randomness and it's at least possible that ten days shows lower germination than nine days due to the randomness. I think you really need to address why does/response approaches are suitable for these data.

   Worse yet, the arbitrary adjustment of the "End" data leads to inappropriate models being fitted to most of the data. As a simple example, take Grouping A at 18.75°C (my Fig. 1). No seeds germinate until day 15, when three seeds germinate. Two days later, three more seeds germinate. The next day one more seed germinates, and that's it; at the end of the experiment seven out of 25 seeds germinate. However, because `G` and `End` in data.frame `dat1` have been set to `(25-7=18)` and `Inf` respectively the model goes on to predict that by day 81 50% of the seeds will have germinated. By the end of the experiment no seeds have germinated for 26 days, but the model predicts

that given another 38 days another 5.5 will. You then go on to use that extrapolated value of 81 days as a known value without error in a regression analysis. Only four out of ten experiments with species A ever achieve 0.5 germination, and yet you have seven data points for `t50` in Figure 3. Figure 3 has seven data points displayed for `t90` but pooling species A and B only three out of 18 experiments ever got that high. Figure 2 is truncated at 43 days, so it looks like the curves are fitted to the data shown, but they are not.

4) Calculating the rates is not quite correct. In the manuscript you state (P8:L13–14) "The inverse of the time is then calculated to obtain the germination rates of each temperature treatment ..." i.e. $R_i = 1/D_i$. This is linearly proportional to the average rate up to that point, but underestimates the actual rate by the decile. For example, if it takes 10 days to get to 0.5 germination, then the germination rate is 1.25/day, not 0.1/day, i.e. $R_i = d * N/D_i$ You can see the effect of not multiplying N by d in table 2, where for each decile the rate goes down, even though in most cases the germination rate is increasing as time goes on. This assumes a constant rate, which of course is not correct, but does allow comparisons, and as long as you only compare the same decile across treatments the error cancels out effectively, but should probably be corrected.

5) I think perhaps step 3a is where one reviewer got concerned. Ignoring the fact that the "data" this analysis is based on are highly suspect, you don't explicitly state why you think a segmented fit is proper, other than it simplifies estimating T_b, T_o and T_c. Figure 3 make the goodness-of-fit appear reasonable, but you don't show the $G = 0$s and you don't extrapolate the lines down to zero so readers can visualize the x intercepts that define T_b and T_c. Given that the `t50` data are largely imaginary, if you extrapolated down to $G = 0$ to get those values T_c looks to be approximately 40°C or higher. So, how good are the segmented fits? You could compare to a quadratic Poisson or negative binomial regression (or better yet GAM) and set T_b and T_c as arbitrarily small non-zero values.

All-in-all, I think given the arbitrariness of the process and the use of gross model extrapolations from the dose/response models as data the method is invalid.

R code

1) I agree with your assessment that the script tries to combine a primer in R with an actual application. This I think is a bad idea; MEE readers generally already know how to use R.

2) The approach adopted in your appendix is quite odd. First, it is obviously designed to be cut-and-pasted into R rather than sourced. This is not an approach I would encourage. Second, it attempts to explain trivial things while never explaining truly unfathomable things like `ddply`.

3) The code is massively Wickhamian. Hadley Whickham is the devil, and if you want a happy and productive life you should eschew his teachings. `ddply` is God's way of telling you your data are improperly structured. Requiring `plyr` and `dplyr` loads an immense number of packages you don't want or need, and produces massive bloat and

obfuscation. This is clearly a conservative, old-school opinion, but I would seriously consider simplifying the R code and reducing the number of required packages.

4) It serves no purpose to have two data.frames (`dat` and `dat1`) of almost identical information. It would have been simple enough to add the required new columns to `dat` and save yourself a lot of trouble. The primary difference between `dat` and `dat1` is that in the first case germination `G` is cumulative, and in the second case event-wise occurrence. The fact that both columns are named `G` is ill-advised.

   As a possible alternative, consider the following:

```
reformat <- function (df)
{
    if (!'Grouping' %in% names(df)) stop
        'the dataframe does not have the required column Grouping'
    if (!'Treatment' %in% names(df)) stop
        'the dataframe does not have the required column Treatment'
    df$Grouping <- factor(df$Grouping)
    df$Treatment <- factor(df$Treatment)
    groups <- levels(df$Grouping)
    treats <- levels(factor(df$Treatment))
    res <- list()
    for (i in groups) {
        for (j in treats) {
            pnt <- df$Grouping==i & df$Treatment ==j
            if (any(pnt==TRUE)) {
                dataset <- paste(i,j,sep='_')
                sub <- df[pnt,3:6]
                names(sub)[3] <- 'Gcum'
                sub$G <- c(0,diff(sub$Gcum))
                sub$G[nrow(sub)] <- max(sub$PG)-max(sub$Gcum)
                sub$start <- sub$Time
                sub$end <- c(sub$start[-1],Inf)
                row.names(sub) <- as.character(1:nrow(sub))
                res[[dataset]] <- sub
            }
        }
    }
    res
}
dat <- read.table('S1 Example Data.txt',header=TRUE)
dat <- reformat(dat)
```

   That produces a list where each experiment is a separate list item with names like `A_18.75`.

```
dat$A_18.25
   Dish Time Gcum PG  G start end
1     1    0    0 25  0     0   1
2     1    1    0 25  0     1   2
3     1    2    0 25  0     2   3
4     1    3    0 25  0     3   4
5     1    4    0 25  0     4   5
6     1    5    0 25  0     5   6
7     1    6    0 25  0     6   7
8     1    7    0 25  0     7   8
9     1    8    0 25  0     8   9
10    1    9    0 25  0     9  10
11    1   10    0 25  0    10  11
12    1   11    0 25  0    11  12
13    1   12    0 25  0    12  13
14    1   13    0 25  0    13  14
15    1   14    0 25  0    14  15
16    1   15    3 25  3    15  16
17    1   16    6 25  3    16  17
18    1   17    6 25  0    17  18
19    1   18    7 25  1    18  19
20    1   19    7 25  0    19  20
⋮
31    1   30    7 25  0    30  31
32    1   31    7 25  0    31  32
33    1   32    7 25  0    32  33
34    1   33    7 25  0    33  34
35    1   34    7 25  0    34  35
36    1   35    7 25  0    35  36
37    1   36    7 25  0    36  37
38    1   37    7 25  0    37  38
39    1   38    7 25  0    38  39
40    1   39    7 25  0    39  40
41    1   40    7 25  0    40  41
42    1   41    7 25  0    41  42
43    1   42    7 25  0    42  43
44    1   43    7 25 18    43 Inf
```

5) The line

```
FSD <- ddply(dat, .(Treatment, Dish, Grouping), filter,
    (Time == max(Time)))
```

produces a data.frame with one row for each unique combination of `Treatment, Dish,` and `Grouping`. This data.frame is fed by `ddply` to a function `FGPfun` which calculates the sums of `G` and `PG` for single lines. There is nothing to sum because it's only one row. Presumably, your intent was to combine `Dishes` within `Groupings` and `Treatments` in which case `Dish` should be deleted from the arguments to FSD. Because in the data there is only ever one `Dish`, it doesn't matter, but the code makes no sense.

Over the last couple of years I have developed a reputation for being an enormous asshole based on my review comments. I don't imagine this one will earn me any friends either. You probably want to be judicious in sharing it with your colleagues. However, if they can address my issues successfully (and maybe I'm just ignorant), I think it will be much better.

Ciao, Dave

Dear Emma, Dear Eduardo,

In an attempt to possibly make amends I have given the problem (and paper) some more thought and contributed some ideas you may find helpful. It's a fascinating problem and interesting data set, so I have enjoyed myself.

Standardization of Germination Data and Reports

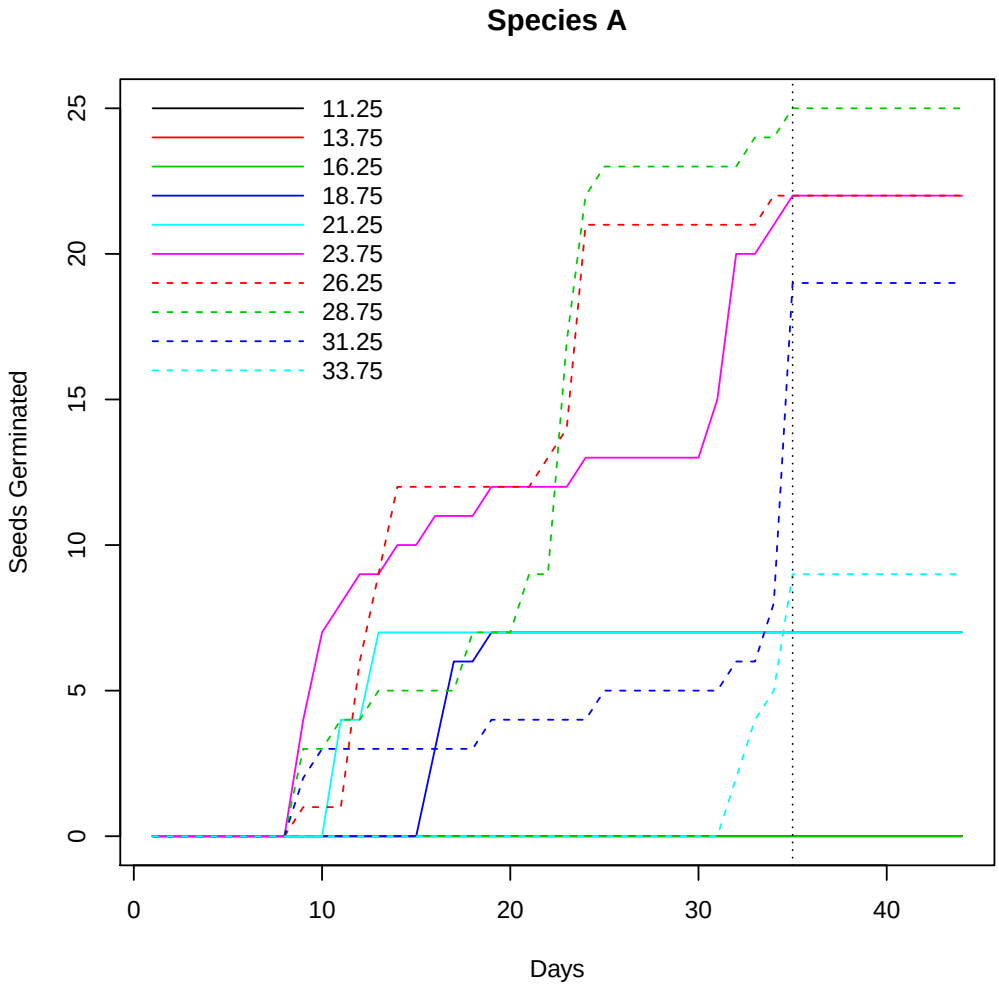
It seems to me that perhaps there are two primary objectives: (1) to establish a standard format for recording seed germination trials to facilitate comparative analyses, and (2) to establish a standard algorithm for establishing T_b, T_o and T_c from such data. Let me come back to objective one later, after examining the data in hand and wrestling with the phenomenon of germination itself. I am not a "traits" ecologist nor a plant ecophysiologist, so my approach might be naíve. Let's start with a simple graphical analysis of the data for species A.

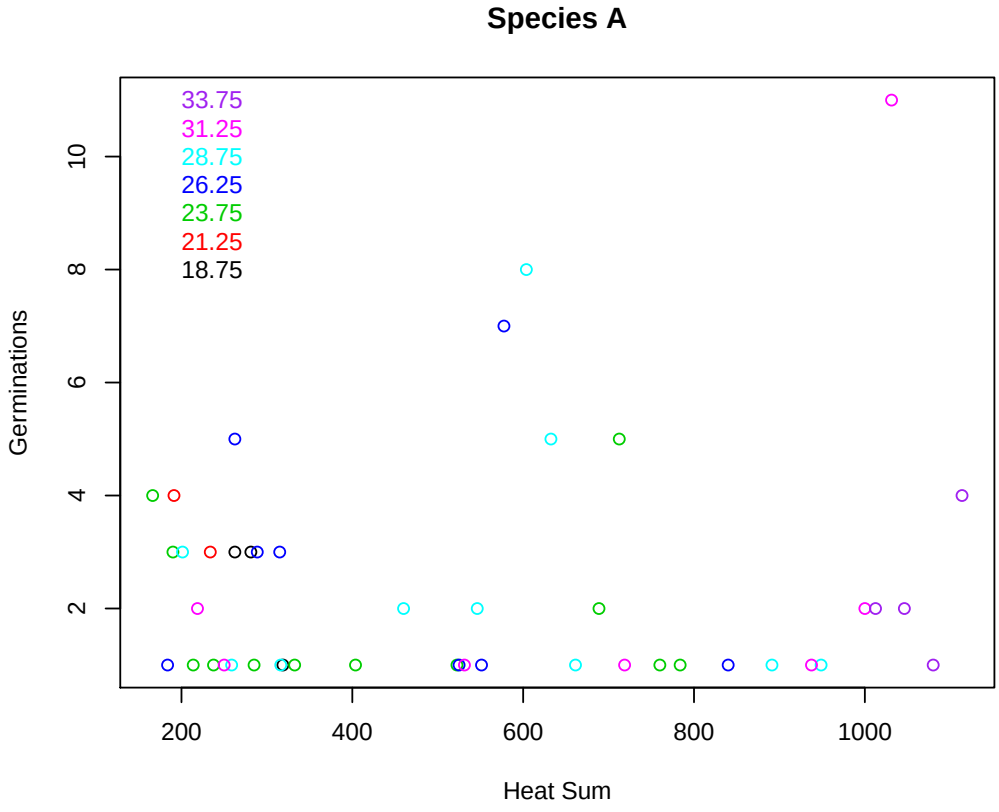
Heat Sum *vs* Threshold Heat

The variation in germination of individual seeds could have at least two possible explanations: (1) it could be a heat-sum requirement where accumulated heat (time $\times$ heat) must meet some threshold, or (2) it could be that seeds have a temperature threshold that has high genetic variability where some seeds react to much lower temperatures than do others. Teasing this apart is a little tricky.

Figure 1 (next page) shows the cumulative germination rates by treatment. Regardless of temperature, seeds needed at least seven days of treatment to germinate. If the trigger was a heat sum, we would have expected the warmer treatments to germinate first (assuming seeds randomly assigned to treatments). Instead, what we see is at temperatures below 18.75°C seeds never germinated regardless of accumulated time. Seeds at temperatures from 21.25°C to 31.25°C all germinated first in a window of days from seven to nine days, followed by 18.75°C at fifteen days and 33.75°C at 30 days. No seeds germinated after 34 days regardless of temperature. Clearly, initial germination is modal in response to heat sum.

Clearly, total germination is also modal with respect to temperature with a peak at 28.75°C . The response is non-symmetric, however, falling off much faster at high temperatures than lower. Since I don't know what the pre-treatment of seeds was, the results seem mixed. If seeds were stored in the dark, it may be that there is a photoperiod threshold, but that extremely low or high temperatures impede germination. Within a specific treatment germinations appears to continue for about 27 days, except for 18.75°C which had a short window of only four days. This in indicative of a fair amount of genetic variability among seeds. It is certainly not a straight-forward heat sum phenomenon, and probability of germination appears to be a model function of temperature.


Species A
11.25
13.75
16.25
18.75
21.25
23.75
26.25
28.75
31.25
33.75
Seeds Germinated
0
5
10
15
20
25
0
10
20
30
40
Days


Species A
33.75
31.25
28.75
26.25
23.75
21.25
18.75
Germinations
2
4
6
8
10
200
400
600
800
1000
Heat Sum

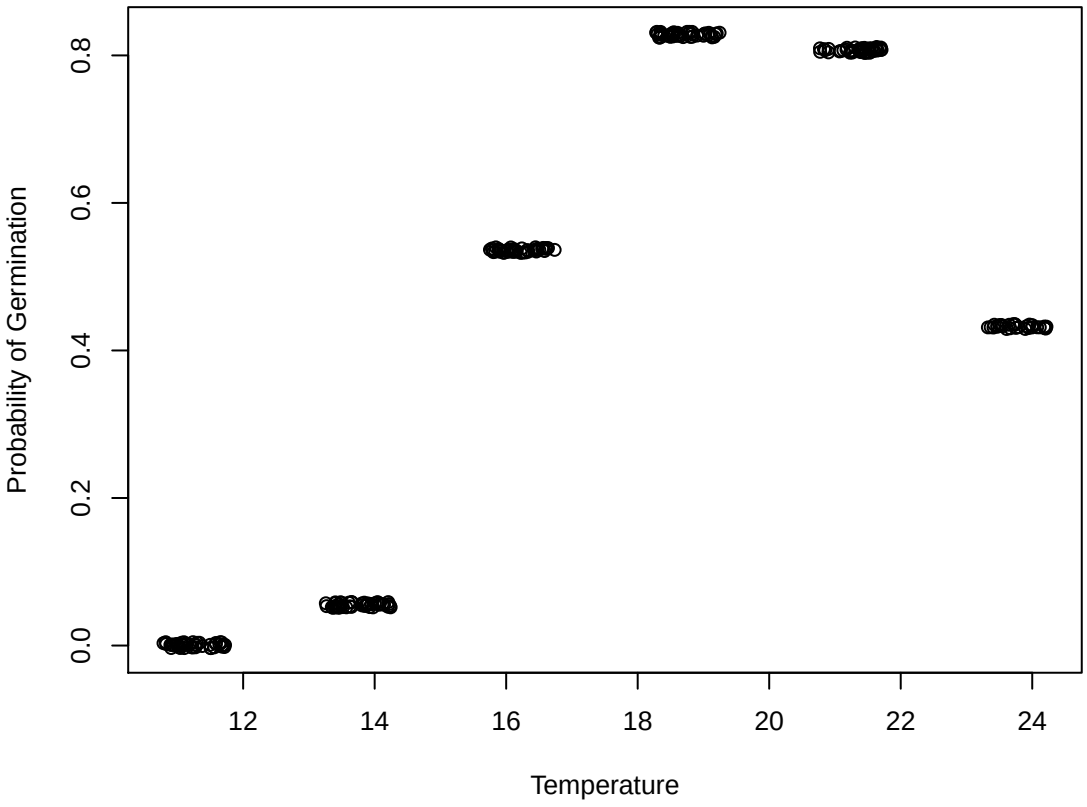
Statistical Considerations and Analyses

The design of the experiment constrains the analysis to some extent. It would be relatively straight forward to treat each seed as independent and fit a quadratic logistic regression to germination status *vs* treatment. We can do that with

```
dat$G[dat$Time==43&dat$Grouping=='A']
[1]  0  0  0  7  7 22 22 25 19  9
master.germ <- c(rep(0,75),rep(1,7),rep(0,18),rep(1,7),rep(0,18),
rep(1,22),rep(0,3),rep(1,22),rep(0,3),rep(1,25),rep(1,19),rep(0,6),
rep(1,9),rep(0,16))
temps <- rep(11.25,25)
for (i in seq(13.75,33.75,2.5)) temps <- c(temps,rep(i,25))
individ.seed.glm <- glm(master.germ temps + I(temps^2), fam='binomial')
Call:
glm(formula = master.germ   temps + I(temps^2), family = "binomial")
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.12473  -0.51555  -0.00248   0.62933   2.04151
Coefficients:
             Estimate Std.  Error z value Pr(>|z|)
(Intercept) -41.25031    6.15509  -6.702 2.06e-11 ***
temps         3.20282    0.47740   6.709 1.96e-11 ***
I(temps^2)   -0.05903    0.00894  -6.603 4.02e-11 ***
---
Signif.  codes:    0 '***' 0.001 '**' 0.01 '*' 0.05 '.'  0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance:  343.43  on 249  degrees of freedom
Residual deviance:  174.84  on 247  degrees of freedom
AIC: 180.84
Number of Fisher Scoring iterations:  7
```

Figure 3 (next page) shows the (jittered) results and it's consistent with what we would expect. However, assuming that the seeds are independent ignores possible dish effects and within-dish chemical communication by seeds (e.g ethylene production). So it's obviously extremely pseudo-replicated.

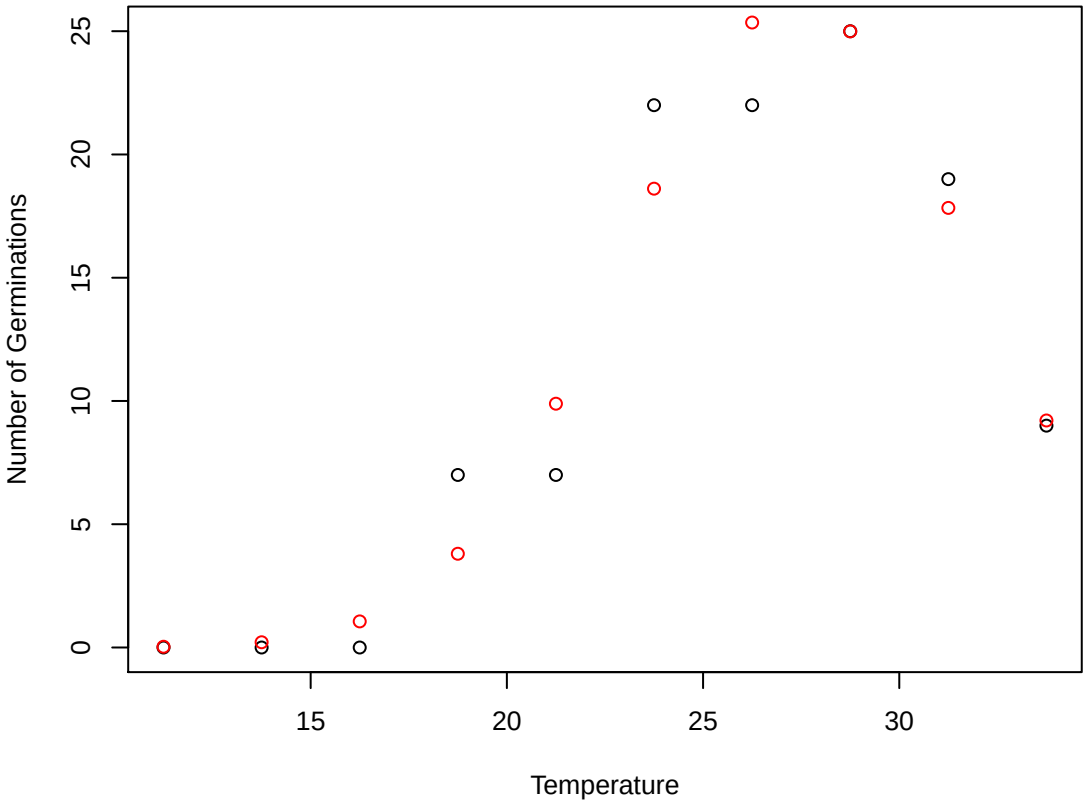
Alternatively, we could treat each dish as a sample unit and model the number of seeds germinated as a Poisson or negative binomial as appropriate. There are too few data points to fit a negative binomial, so we can try a quadratic Poisson.




```
alt.temps <- seq(11.25,33.75,2.5)
comb.germ <- glm(germs alt.temps + I(alt.temps^2),family='poisson')
summary(comb.germ)
Call:
glm(formula = germs   alt.temps + I(alt.temps^2), family = "poisson")
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4550  -0.6744  -0.1593   0.2052   1.4654
Coefficients:
               Estimate Std.  Error z value Pr(>|z|)
(Intercept)    -16.13710    3.20254  -5.039 4.68e-07 ***
alt.temps        1.41685    0.24482   5.787 7.15e-09 ***
I(alt.temps^2)  -0.02586    0.00462  -5.598 2.17e-08 ***
---
Signif.  codes:    0 '***' 0.001 '**' 0.01 '*' 0.05 '.'  0.1 ' ' 1
(Dispersion parameter for poisson family taken to be 1)
```

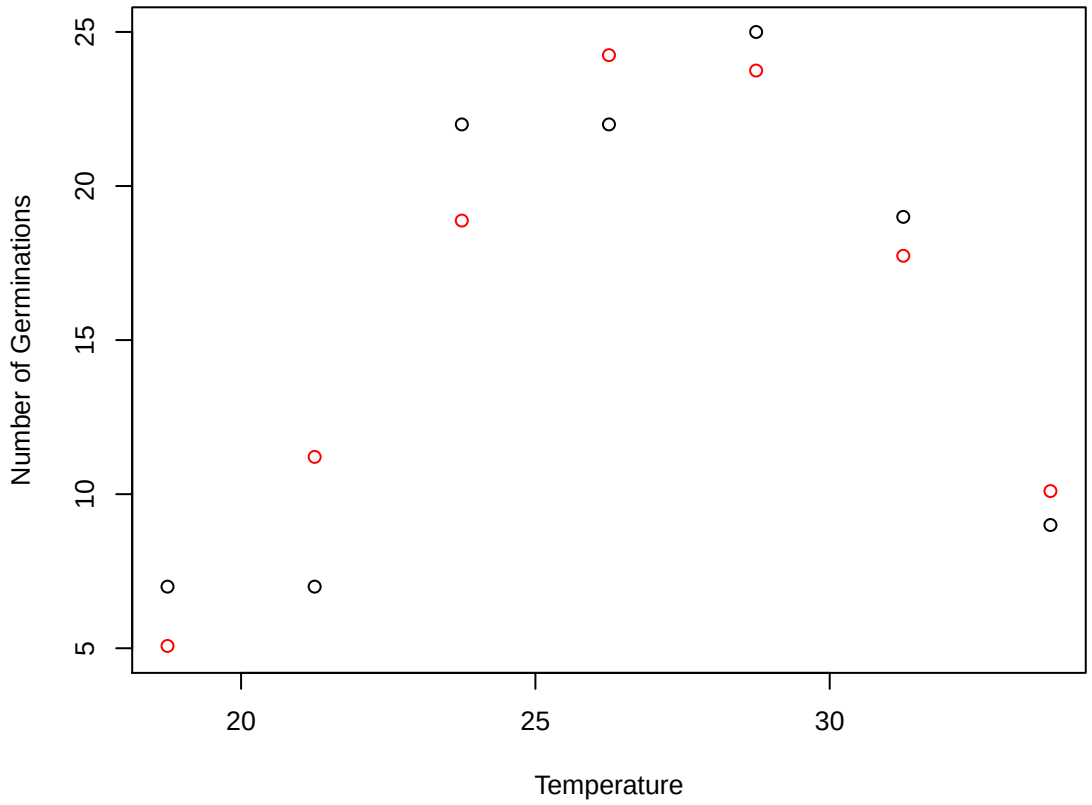
```
    Null deviance:  104.5379  on 9  degrees of freedom
Residual deviance:      6.8203  on 7  degrees of freedom
AIC: 44.217
Number of Fisher Scoring iterations:  5
```

That's an astonishing reduction in null deviance. Figure 4 shows the results. Unfortunately, the fitted values exceed 25, which is not possible, so we're still in statistical limbo. If we limit the analysis to non-zero cases (quasi zero-inflated analysis) the problem goes away.




```
alt.alt.temps <- glm(germs[4:10] alt.temps[4:10]+
    I(alt.temps[4:10]^2),fam='poisson')
summary(alt.alt.temps)
Call:
glm(formula = germs[4:10]   alt.temps[4:10] + I(alt.temps[4:10]^2),
    family = "poisson")
Deviance Residuals:
      1        2        3        4        5        6        7
 0.8075  -1.3517   0.6995  -0.4640   0.2545   0.2965  -0.3530
Coefficients:
```

```
                       Estimate Std.  Error z value Pr(>|z|)
(Intercept)          -12.959371   3.749952  -3.456 0.000549 ***
alt.temps[4:10]        1.184418   0.283972   4.171 3.03e-05 ***
I(alt.temps[4:10]^2)  -0.021686   0.005297  -4.094 4.24e-05 ***
---
Signif.  codes:    0 '***' 0.001 '**' 0.01 '*' 0.05 '.'  0.1 ' ' 1
(Dispersion parameter for poisson family taken to be 1)
    Null deviance:  25.3561  on 6  degrees of freedom
Residual deviance:    3.4612  on 4  degrees of freedom
AIC: 40.857
Number of Fisher Scoring iterations:  4
```



You would have to convince a reviewer that deleting the zero cases was legit, but I think that's doable. If we solve for zero to find T_b and T_C we get 15.14°C and 39.48°C with T_o of 24.85 germinations at 27.31°C .

Segmented Solution

Given the customizations required in the previous fit, you might still prefer the segmented solution you used before. The primary problem (from my perspective) was using dose/response methodology on a time series and then using vastly extrapolated predictions

as data. There would seem to be other ways to get a segmented fit. However, as I'm sure you know, it's tricky. First, the germinations could show numerous zeros at the beginning or end of the series if the range of temperatures is large. Alternatively, the lowest and highest temperatures could be inside the critical temperatures so that no zeros occur. In addition, the curve will not necessarily be strictly modal, and zeros could occur in the interior of non-zero points. Here's my proposal

```
1) Make sure that germination occurred for at least one temperature.
2) Find which temperature in the series produced maximum germination = Xo
   If there are ties, find the mean of the first and last cases = Xo
3) If the sequence of germinations begins with multiple zeros, find the
   last zero before the non-zero values, else use the first point = Xb
4) If the sequence ends with zeros, find the first zero after the last
    non-zero value, else use the last point = Xc
5) fit two linear regressions from Xb to Xo and Xo to Xc
6) solve for the X intercepts of the two regression = Tb and Tc
7) Find the intersection of the regression lines = To
```

I haven't tested it really fully yet, but it seems to work.

```
plot.crits <- function (germ,temps)
{
    if (length(germ) != length(temps))
        stop('vectors germ and temps are different length')

    numcases <- length(germ)

    if (!any(germ>0))
        stop('No non-zero cases to analyze')

    maxgerm <- max(germ)
    left <- min(seq(1,numcases)[germ==maxgerm])
    right <- max(seq(1,numcases)[germ==maxgerm])

    midx <- mean(c(left,right))

    posi <- seq(1:numcases)[germ > 0]
    minx <- max(1,min(posi)-1)
    maxx <- min(max(posi)+1,numcases)

    if (midx > minx) {
        tmpx <- temps[minx:floor(midx)]
        tmpy <- germ[minx:floor(midx)]
        x1 <- lm(tmpy tmpx)$coefficients
```

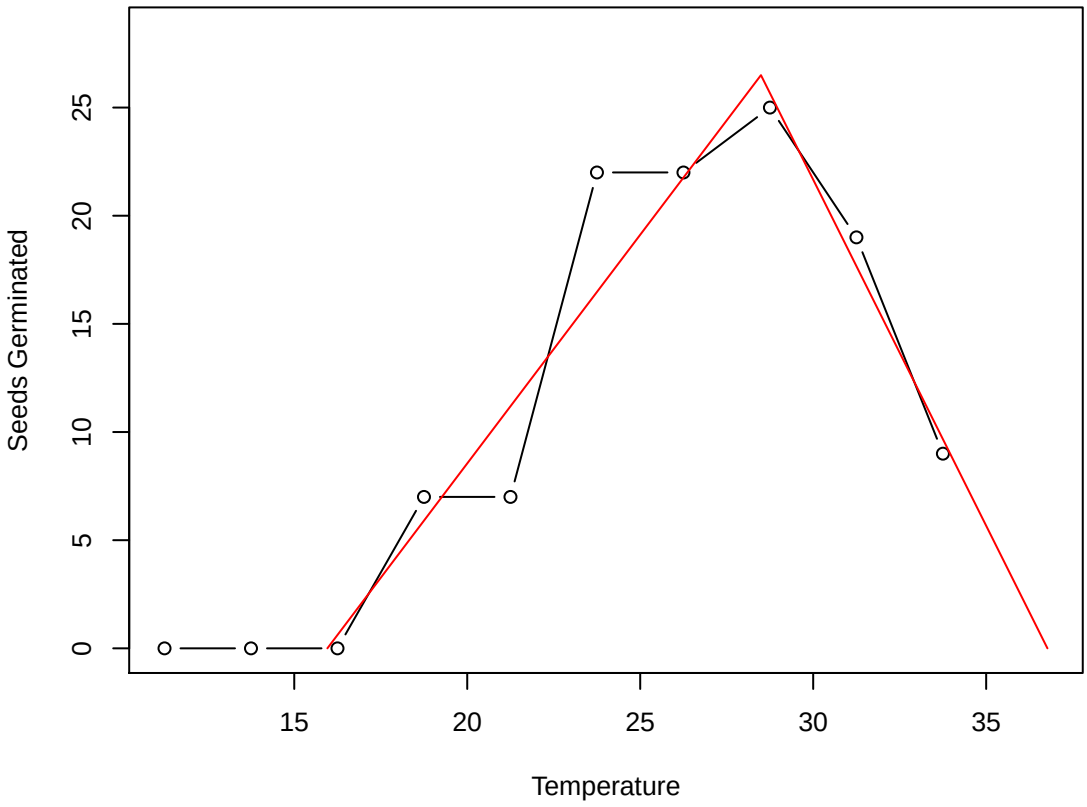
```
        tb <- -x1[1]/x1[2]
    } else {
        tb <- germ[minx]
    }

    if (maxx > midx) {
        tmpx <- temps[ceiling(midx):maxx]
        tmpy <- germ[ceiling(midx):maxx]
        x2 <- lm(tmpy tmpx)$coefficients
        tc <- -x2[1]/x2[2]
    } else {
        tc <- maxx
    }

    if (midx > minx && maxx > midx) {
        to <- (x2[1]-x1[1])/(x1[2]-x2[2])
        maxg <- x1[1]+x1[2]*to
    } else {
        to <- midx
        maxg <- germ[midx]
    }

    plot(temps,germ,type='b',xlab='Temperature',ylab='Seeds Germinated',
        xlim=c(min(temps,tb),max(temps,tc)),ylim=c(0,max(germ,to)))
    lines(c(tb,to,tc),c(0,maxg,0),col=2)
    cat(paste(' Tb = ',round(tb,3),'\n','To = ',round(to,3),
        '\n','Tc = ',round(tc,3),'\n'))
}
```

Here's an example (next page) applied to Species A.




```
Tb =  15.957
To =  28.49
Tc =  36.771
```

Anyway, I hope there are some helpful suggestions in here.

Ciao, Dave