

NicheMapR and DEB theory I: the microclimate and ectotherm models

Michael Kearney

2023-06-09

Contents

Overview	1
Installing NicheMapR and associated packages	1
Run the microclimate model for the mean day of each month	2
Plot the microclimate model output	2
Run the microclimate model with the gridmet historical daily weather data for 2017-2020	7
Plot the microclimate model output	8
Run the ectotherm model to compute the heat budget	10
Plot the ectotherm model results	11
References	13

Overview

This document runs simulations illustrating the microclimate model and ectotherm model functions of the NicheMapR package. Another document (part II) illustrates the DEB functions of the NicheMapR package including their integration with the ectotherm model.

Installing NicheMapR and associated packages

Below is the code to install the latest version of NicheMapR directly from github, as well as the microclima package (see Maclean et al. 2019) which has some complementary functions for computing microclimates. It also installs the RNCEP package (Kemp et al. 2012) which provides 6-hourly historical input weather data estimates for anywhere in the world, as well as the Climate Research Unit's (CRU) monthly global climate data (New et al. 2002). You may also install previous non-beta releases of NicheMapR from <https://github.com/mrke/NicheMapR/releases> Note that to install NicheMapR from github on a Windows machine you will also need to have Rtools installed, which can be found at <https://cran.r-project.org/bin/windows/Rtools/>

To install on a mac you will need to make sure you have a Fortran compiler installed. See https://mrke.github.io/getting_started/ for more details on installation as well as links to Fortran compilers for macs.

```
install.packages("devtools")
install.packages("RNCEP")
library(devtools) # load the devtools package
install_github("mrke/NicheMapR")
install_github("ilyamaclean/microclima")
library(NicheMapR) # load the NicheMapR package
get.global.climate() # this will download and unzip a 0.5 gig global monthly climate data
```

Run the microclimate model for the mean day of each month

Run the microclimate model for Baton Rouge from the global monthly climate, with all default settings and extract the results.

```
# load packages
library(NicheMapR)
library(knitr) # this is to make nice-looking tables within the R Markdown document

loc <- c(-91.18, 30.449) # set location - decimal degrees 'c(long, lat)'

# call the microclimate model, global climate database
# implementation
micro <- micro_global(loc = loc)

metout <- as.data.frame(micro$metout) # above ground microclimatic conditions, min shade
shadmet <- as.data.frame(micro$shadmet) # above ground microclimatic conditions, max shade
soil <- as.data.frame(micro$soil) # soil temperatures, minimum shade
shadsoil <- as.data.frame(micro$shadsoil) # soil temperatures, maximum shade

# append mock dates
days <- rep(seq(1, 12), 24)
days <- days[order(days)]
dates <- days + metout$TIME/60/24 - 1 # dates for hourly output
dates2 <- seq(1, 12, 1) # dates for daily output
plotmetout <- cbind(dates, metout)
plotsoil <- cbind(dates, soil)
plotshadmet <- cbind(dates, shadmet)
plotshadsoil <- cbind(dates, shadsoil)

# extract shade values for plot labelling
minshade <- micro$minshade
maxshade <- micro$maxshade
```

Plot the microclimate model output

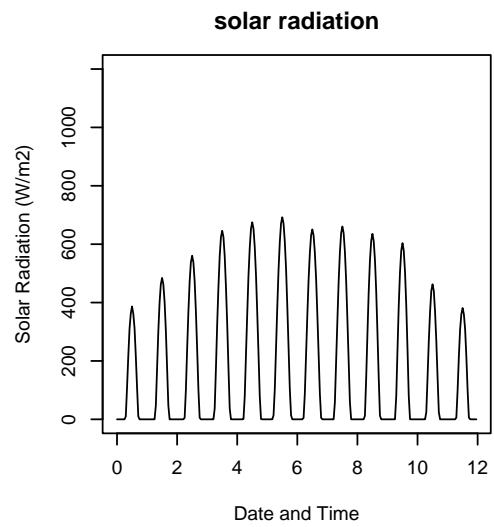
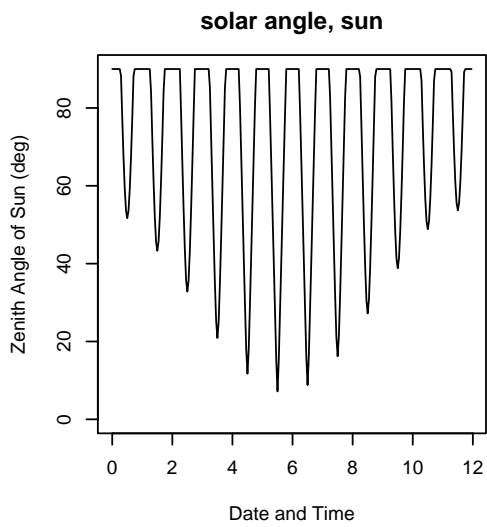
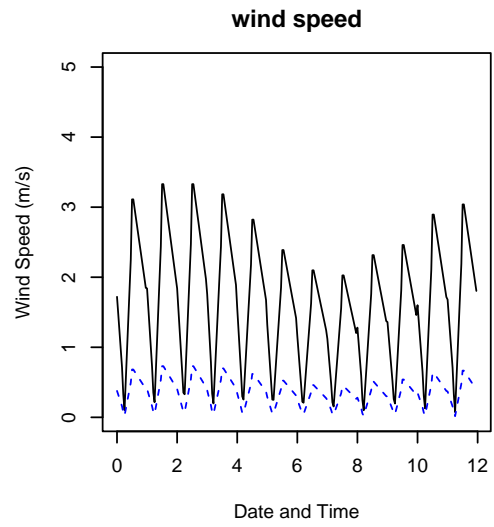
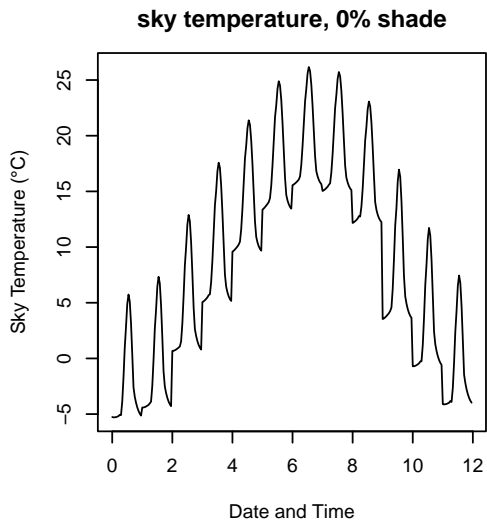
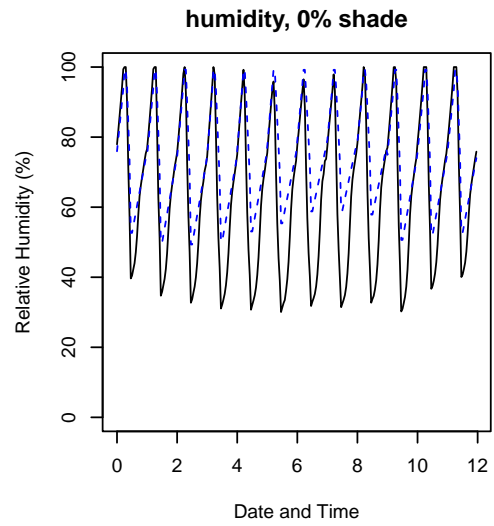
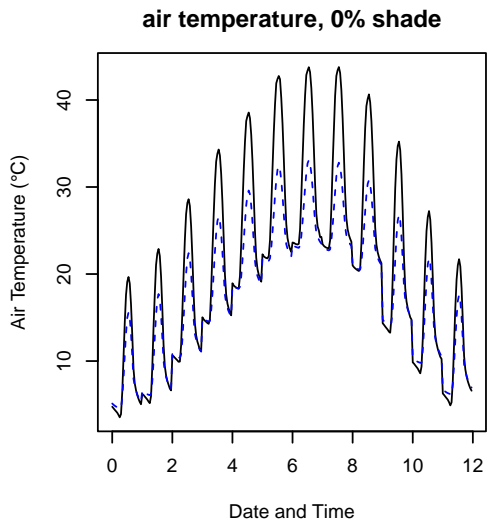
The following code plots the aboveground conditions under the minimum shade scenario (full sun).

```
# plotting above-ground conditions in minimum shade
par(mfrow = c(3, 2)) # three rows of two columns
par(oma = c(2, 1, 1, 2) + 1) # margin spacing
par(mar = c(2, 2, 1, 1) + 2) # margin spacing
par(mgp = c(3, 1, 0)) # margin spacing
```

```

with(plotmetout, {
  plot(TALOC ~ dates, xlab = "Date and Time", ylab = "Air Temperature (°C)",
       type = "l", main = paste("air temperature, ", minshade[1],
                                "% shade", sep = ""))
})
with(plotmetout, {
  points(TAREF ~ dates, xlab = "Date and Time", ylab = "Air Temperature (°C)",
        type = "l", lty = 2, col = "blue")
})
with(plotmetout, {
  plot(RHLOC ~ dates, xlab = "Date and Time", ylab = "Relative Humidity (%)",
       type = "l", ylim = c(0, 100), main = paste("humidity, ",
                                                    minshade[1], "% shade", sep = ""))
})
with(plotmetout, {
  points(RH ~ dates, xlab = "Date and Time", ylab = "Relative Humidity (%)",
        type = "l", col = "blue", lty = 2, ylim = c(0, 100))
})
with(plotmetout, {
  plot(TSKYC ~ dates, xlab = "Date and Time", ylab = "Sky Temperature (°C)",
       type = "l", main = paste("sky temperature, ", minshade[1],
                                "% shade", sep = ""))
})
with(plotmetout, {
  plot(VREF ~ dates, xlab = "Date and Time", ylab = "Wind Speed (m/s)",
       type = "l", main = "wind speed", ylim = c(0, 5))
})
with(plotmetout, {
  points(VLOC ~ dates, xlab = "Date and Time", ylab = "Wind Speed (m/s)",
        type = "l", lty = 2, col = "blue")
})
with(plotmetout, {
  plot(ZEN ~ dates, xlab = "Date and Time", ylab = "Zenith Angle of Sun (deg)",
       type = "l", main = "solar angle, sun", ylim = c(0, 90))
})
with(plotmetout, {
  plot(SOLR ~ dates, xlab = "Date and Time", ylab = "Solar Radiation (W/m2)",
       type = "l", main = "solar radiation", ylim = c(0, 1200))
})

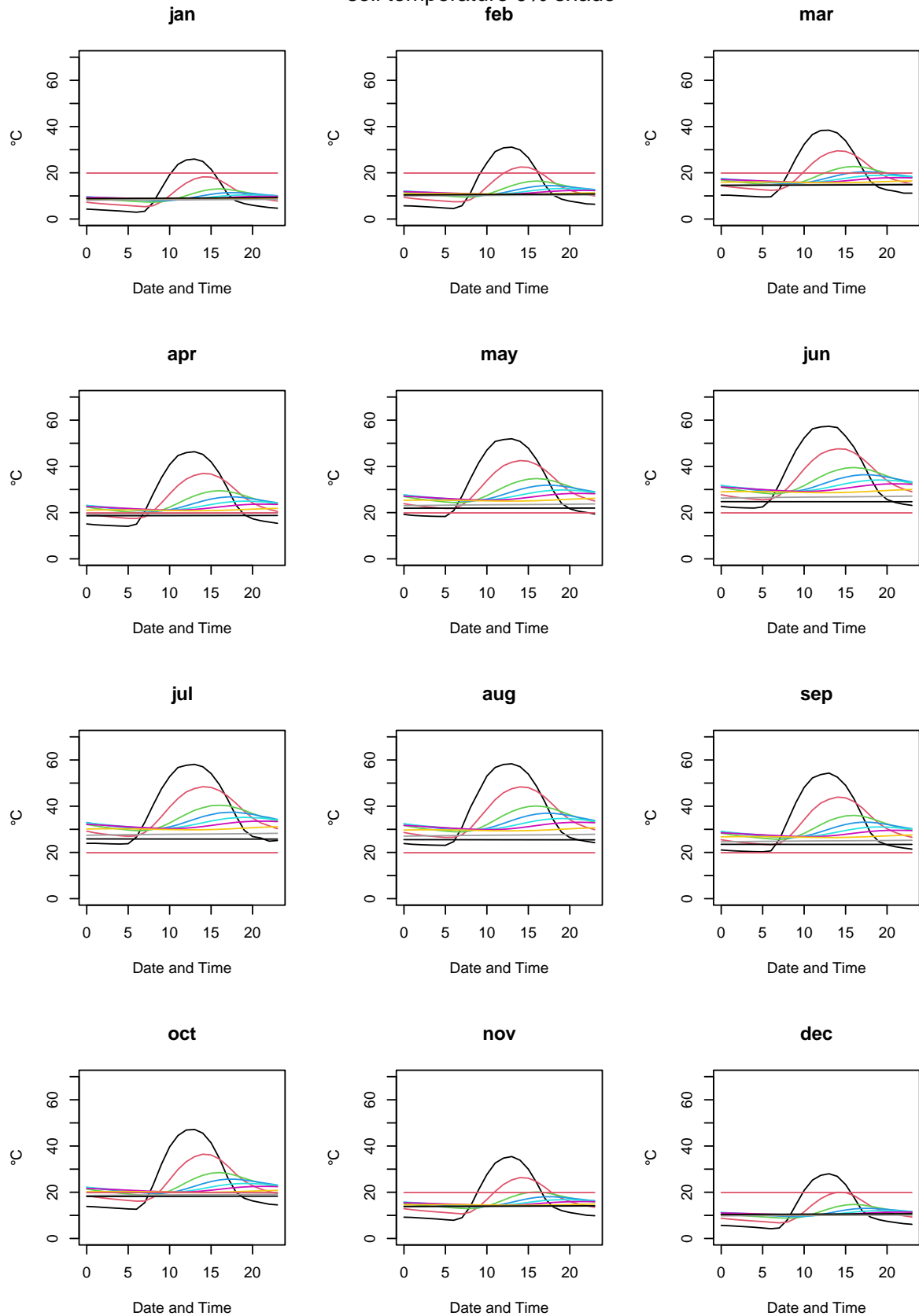
```



The next code plots daily soil temperature for middle day of each at each month for minimum shade.

```
par(mfrow = c(4, 3)) # three four of 1 columns
months <- c("jan", "feb", "mar", "apr", "may", "jun", "jul",
  "aug", "sep", "oct", "nov", "dec")
DOYS <- unique(metout$DOY)
for (j in 1:12) {
  plotsoil <- subset(soil, DOY == DOYS[j])
  for (i in 1:10) {
    if (i == 1) {
      plot(plotsoil[, 2]/60, plotsoil[, i + 2], xlab = "Date and Time",
        ylab = "°C", col = i, type = "l", ylim = c(0,
          70), main = months[j])
    } else {
      points(plotsoil[, 2]/60, plotsoil[, i + 2], xlab = "Date and Time",
        ylab = "Soil Temperature
        (°C)", col = i,
        type = "l")
    }
  }
}
mtext(paste0("soil temperature ", minshade[1], "% shade"), side = 3,
  line = -1.5, outer = TRUE)
```

soil temperature 0% shade



Run the microclimate model with the gridmet historical daily weather data for 2017-2020

This section runs the microclimate model using the gridmet weather data for the USA. You can also use the global NCEP product, which is 6-hourly global 2.5 degree (~250 km) resolution data set of a suite of atmospheric variables, or ERA5 which is a . The data can be queried via the R package RNCEP (Kemp et al. 2012). We have developed scripts to disaggregate the NCEP data to hourly and downscale it to a finer spatial resolution using the ‘microclima’ package together with the ‘elevatr’ package (Kearney et al. 2020). This adjustment includes elevation effects (via adiabatic lapse rates), coastal effects, cold air drainage and topographic effects (e.g. hill shade). The main inaccuracies will be due to rainfall, which remains at the 250 km resolution (but a vector of local hourly rainfall can be passed to the function and it will use that rainfall instead).

The following code runs the ‘micro_ncep’ function for Baton Rouge over the years 2017 to 2020. It will take a while to download all the NCEP data (you can download the data locally and run it from that, specifying the folder it is in via the argument ‘spatial’, to speed things up).

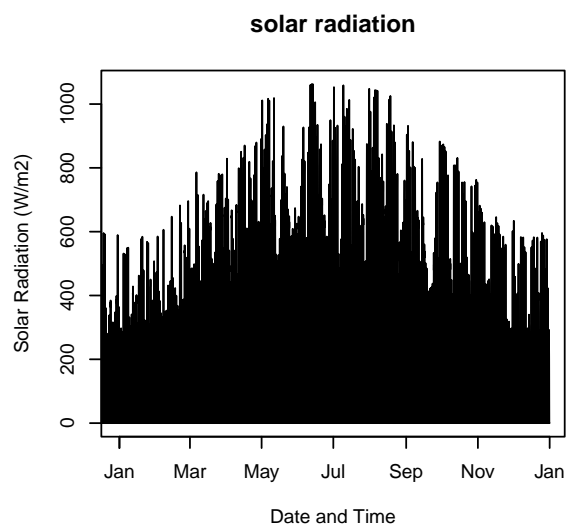
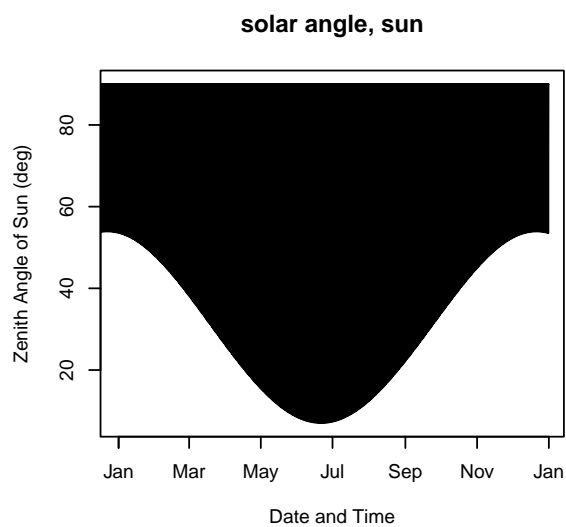
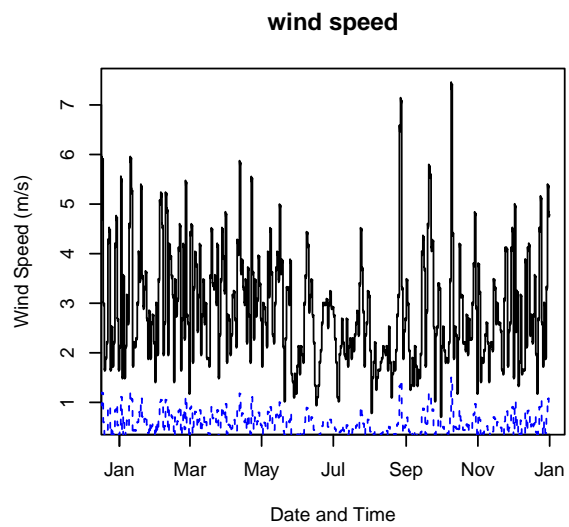
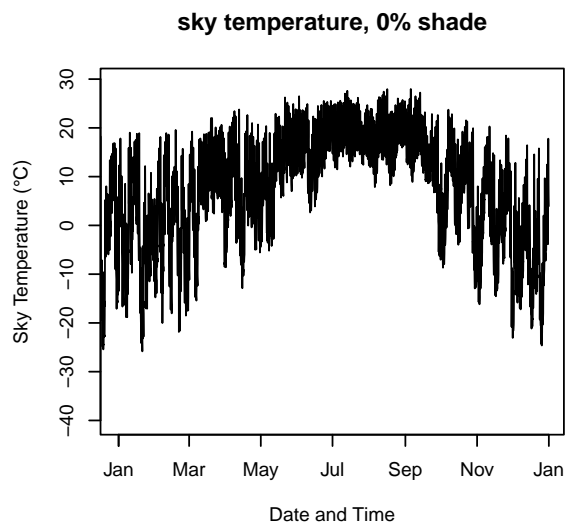
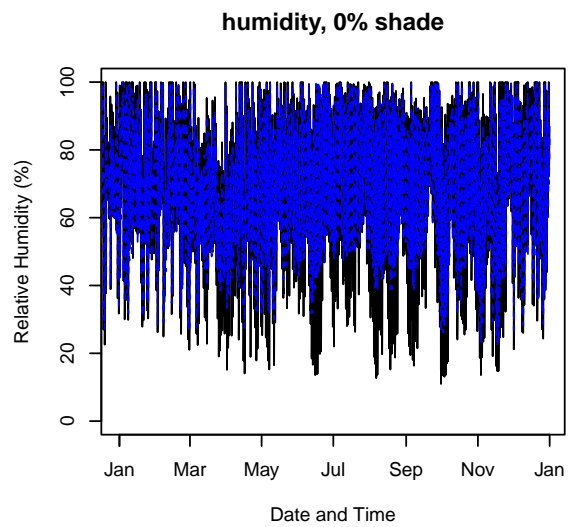
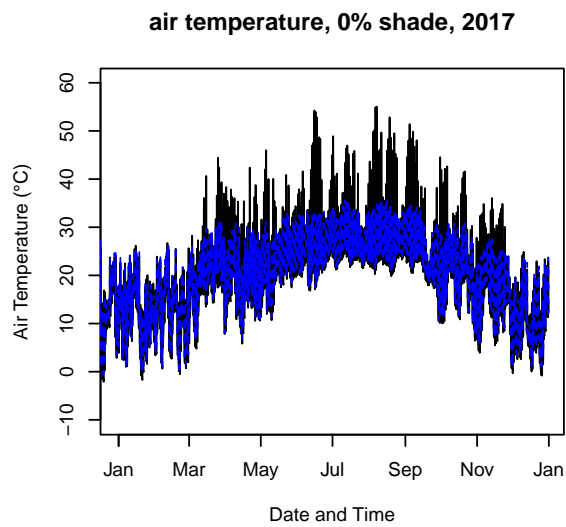
```
loc <- c(-91.18, 30.449) # set location - decimal degrees 'c(long, lat)'  
  
# call the microclimate model, global climate database  
# implementation  
dstart <- "01/01/2017"  
dfinish <- "31/12/2020"  
  
# set depths and height to simulate  
DEP <- c(0, 2.5, 5, 10, 15, 20, 30, 50, 100, 200) # depths to simulate (cm)  
Usrhyt <- 0.01 # height of midpoint of animal (m)  
  
# run the microclimate model and extract some output for  
# plotting micro <- micro_ncep(loc = loc, dstart = dstart,  
# dfinish = dfinish, DEP = DEP, #runshade = 1, Usrhyt =  
# Usrhyt) micro <- micro_usa(loc = loc, dstart = dstart,  
# dfinish = dfinish, DEP = DEP, #runshade = 1, Usrhyt =  
# Usrhyt)  
  
# save the results for later save(micro, file =  
# 'micro_baton_rouge_2017_2020.Rda')  
load("micro_baton_rouge_2017_2020.Rda") # uncomment this and comment out the two lines above once micr  
  
metout <- as.data.frame(micro$metout) # above ground microclimatic conditions, min shade  
shadmet <- as.data.frame(micro$shadmet) # above ground microclimatic conditions, max shade  
soil <- as.data.frame(micro$soil) # soil temperatures, minimum shade  
shadsoil <- as.data.frame(micro$shadsoil) # soil temperatures, maximum shade  
  
# extract dates  
dates <- micro$dates  
dates2 <- micro$dates2  
plotmetout <- cbind(dates, metout)  
plotsoil <- cbind(dates, soil)  
plotshadmet <- cbind(dates, shadmet)  
plotshadsoil <- cbind(dates, shadsoil)  
  
# extract shade values for plot labelling  
minshade <- micro$minshade[1]
```

```
maxshade <- micro$maxshade[1]
```

Plot the microclimate model output

Plotting the aboveground conditions under the minimum shade scenario (full sun) for 2020.

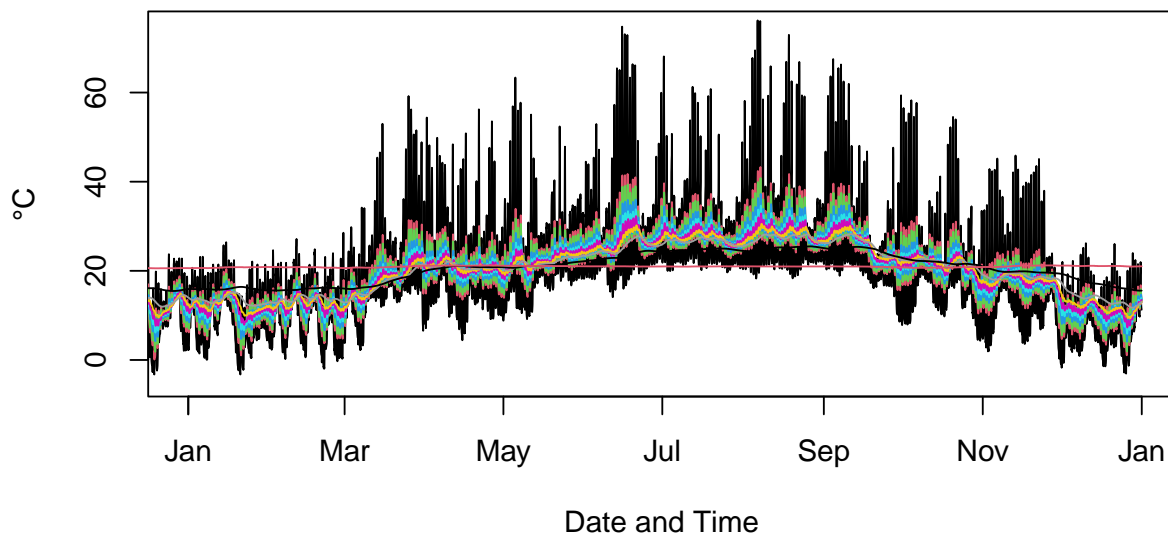
```
par(mfrow = c(3, 2)) # three rows of two columns
# plotting above-ground conditions in minimum shade, 2017
xlim <- c(as.POSIXct("2020-01-01"), as.POSIXct("2020-12-31"))
with(plotmetout, {
  plot(TALOC ~ dates, xlim = xlim, xlab = "Date and Time",
       ylab = "Air Temperature (°C)", type = "l", main = paste("air temperature, ",
       minshade[1], "% shade, 2017", sep = ""))
})
with(plotmetout, {
  points(TAREF ~ dates, xlab = "Date and Time", ylab = "Air Temperature (°C)",
        type = "l", lty = 2, col = "blue")
})
with(plotmetout, {
  plot(RHLOC ~ dates, xlim = xlim, xlab = "Date and Time",
       ylab = "Relative Humidity (%)", type = "l", ylim = c(0,
       100), main = paste("humidity, ", minshade[1], "% shade",
       sep = ""))
})
with(plotmetout, {
  points(RH ~ dates, xlab = "Date and Time", ylab = "Relative Humidity (%)",
        type = "l", col = "blue", lty = 2, ylim = c(0, 100))
})
with(plotmetout, {
  plot(TSKYC ~ dates, xlim = xlim, xlab = "Date and Time",
       ylab = "Sky Temperature (°C)", type = "l", main = paste("sky temperature, ",
       minshade[1], "% shade", sep = ""))
})
with(plotmetout, {
  plot(VREF ~ dates, xlim = xlim, xlab = "Date and Time", ylab = "Wind Speed (m/s)",
       type = "l", main = "wind speed")
})
with(plotmetout, {
  points(VLOC ~ dates, xlab = "Date and Time", ylab = "Wind Speed (m/s)",
        type = "l", lty = 2, col = "blue")
})
with(plotmetout, {
  plot(ZEN ~ dates, xlim = xlim, xlab = "Date and Time", ylab = "Zenith Angle of Sun (deg)",
       type = "l", main = "solar angle, sun")
})
with(plotmetout, {
  plot(SOLR ~ dates, xlim = xlim, xlab = "Date and Time", ylab = "Solar Radiation (W/m2)",
       type = "l", main = "solar radiation")
})
```

Plotting daily soil temperature over the year 2020 at each depth for minimum shade.

```
par(mfrow = c(1, 1)) # three four of 1 columns
for (i in 1:10) {
  if (i == 1) {
    plot(plotsoil[, 1], plotsoil[, i + 3], xlim = xlim, xlab = "Date and Time",
         ylab = "°C", col = i, type = "l", ylim = c(-5, 75))
  } else {
    points(plotsoil[, 1], plotsoil[, i + 3], xlab = "Date and Time",
          ylab = "Soil Temperature
(°C)", col = i, type = "l")
  }
}
mtext(paste0("soil temperature ", minshade[1], "% shade, 2020"),
      side = 3, line = -1.5, outer = TRUE)
```

soil temperature 0% shade, 2020



Run the ectotherm model to compute the heat budget

This last section runs the ectotherm model of NicheMapR for the green anole at the site simulated over 2017 to 2020. The ectotherm model contains the fundamental equations for solving the heat budget, i.e. determining the body temperature that results from a given radiation, convection, conduction and evaporative heat balance. The model includes sophisticated algorithms for behavioural thermoregulation and activity patterns, allowing specification of daily activity phase (nocturnal, diurnal, crepuscular), whether it can seek shade, climb or move underground, and what its body temperature thresholds are for various responses.

First specify some morphological parameters - the shape and solar absorptivity.

```
alpha_min <- 0.8 # minimum solar absorbtivity (dec %)
alpha_max <- 0.85 # maximum solar absorbtivity (dec %)
shape <- 3 # lizard
```

Now specify the thermal physiology and behaviour.

```
Ww_g <- 3.5 # wet weight, grams
T_RB_min <- 15 # min Tb at which they will attempt to leave retreat
T_B_min <- 20 # min Tb at which leaves retreat to bask
T_F_min <- 25 # minimum Tb at which activity occurs
T_F_max <- 35 # maximum Tb at which activity occurs
T_pref <- 33.8 # preferred Tb (will try and regulate to this)
CT_max <- 36 # critical thermal minimum (affects choice of retreat)
CT_min <- 9 # critical thermal maximum (affects choice of retreat)
mindepth <- 2 # min depth (node, 1-10) allowed
maxdepth <- 10 # max depth (node, 1-10) allowed
shade_seek <- 1 # shade seeking?
burrow <- 1 # can it burrow?
climb <- 0 # can it climb to thermoregulate?
nocturn <- 0 # nocturnal activity
crepus <- 0 # crepuscular activity
diurn <- 1 # diurnal activity
```

Run the ectotherm model and retrieve the output.

```
ecto <- ectotherm(Ww_g = Ww_g, alpha_max = alpha_max, alpha_min = alpha_min,
  T_F_max = T_F_max, T_F_min = T_F_min, T_B_min = T_B_min,
  T_RB_min = T_RB_min, CT_max = CT_max, CT_min = CT_min, T_pref = T_pref,
  mindepth = mindepth, maxdepth = maxdepth, shade_seek = shade_seek,
  burrow = burrow, climb = climb, nocturn = nocturn, diurn = diurn,
  crepus = crepus)

# retrieve output
environ <- as.data.frame(ecto$environ) # behaviour, Tb and environment
enbal <- as.data.frame(ecto$enbal) # heat balance outputs
masbal <- as.data.frame(ecto$masbal) # mass balance outputs
environ <- cbind(environ, metout$SOLR) # add solar radiation for activity window plots
colnames(environ)[ncol(environ)] <- "Solar"

# append mock dates
environ <- as.data.frame(cbind(dates, environ))
masbal <- as.data.frame(cbind(dates, masbal))
enbal <- as.data.frame(cbind(dates, enbal))
```

Plot the ectotherm model results

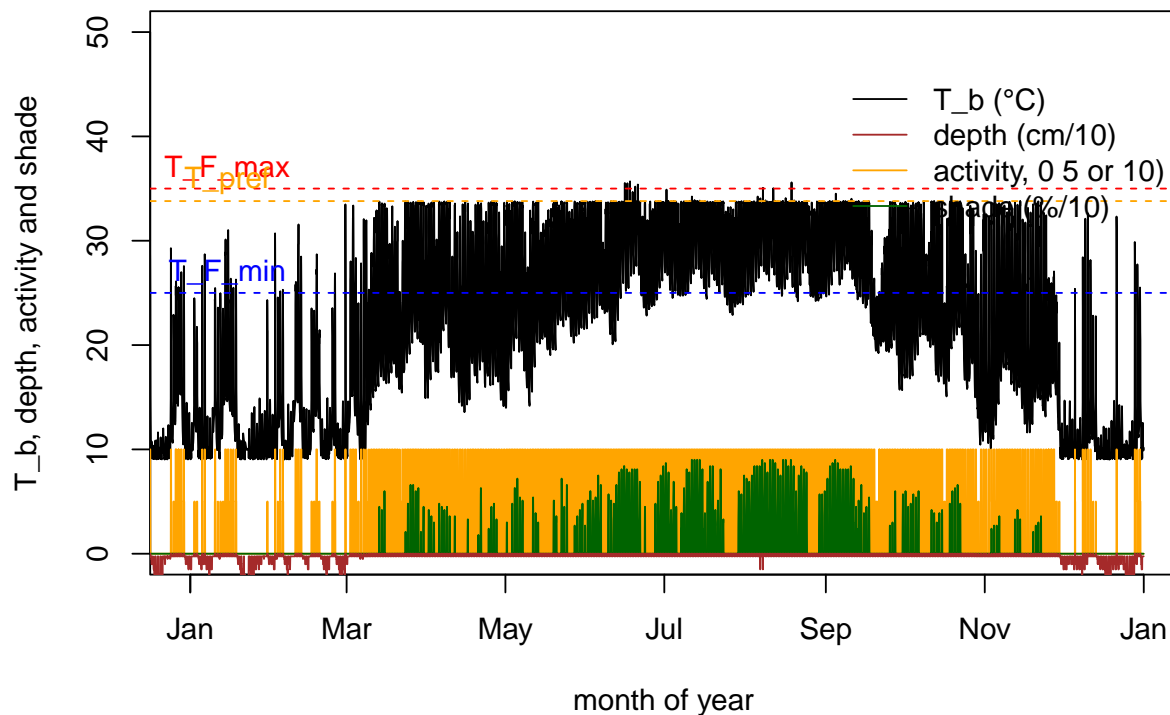
Finally, we can plot the results, just for 2020. The first plot shows the body temperature trace as well as indicating when the animal was active (0 = inactive, 1 = basking, 2 = foraging, multiplied by 5 on the graph to make it clearer), how much shade it needed (divided by 10 for plotting), and how deep it went underground (again divided by 10).

```
par(mfrow = c(1, 1)) # two rows of 1 columns
with(environ, plot(TC ~ dates, xlim = xlim, ylab = "T_b, depth, activity and shade",
  xlab = "month of year", ylim = c(0, 50), type = "l"))
with(environ, points(ACT * 5 ~ dates, type = "l", col = "orange"))
with(environ, points(SHADE/10 ~ dates, type = "l", col = "dark green"))
```

```

with(enviro, points(DEP/10 ~ dates, type = "l", col = "brown"))
abline(T_F_max, 0, lty = 2, col = "red")
abline(T_F_min, 0, lty = 2, col = "blue")
abline(T_pref, 0, lty = 2, col = "orange")
text(xlim[1] + 15 * 3600 * 24, T_F_max + 2, "T_F_max", col = "red")
text(xlim[1] + 15 * 3600 * 24, T_F_min + 2, "T_F_min", col = "blue")
text(xlim[1] + 15 * 3600 * 24, T_pref + 2, "T_pref", col = "orange")
legend(x = xlim[2] - 120 * 3600 * 24, y = T_F_max + 12, legend = c("T_b (°C)",
  "depth (cm/10)", "activity, 0 5 or 10)", "shade (%/10)",
  col = c("black", "brown", "orange", "dark green"), lty = rep(1,
    4), bty = "n")

```



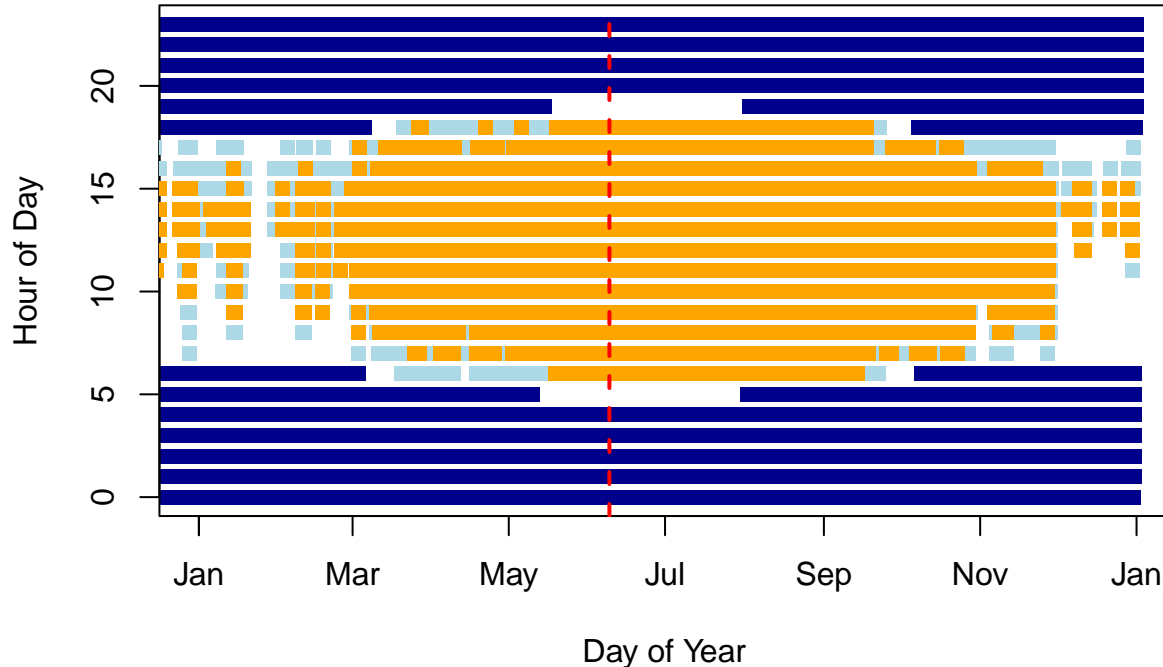
The second plot shows the seasonal activity window, with day of year along the x and time of day along the y. Night hours are indicated in dark blue, and show the photoperiod change through the year. The light blue points show the periods of basking and the gold points show when activity was possible. This workshop (vertical line) is well within the main activity season for this species.

```

# seasonal activity plot (dark blue = night, light blue =
# basking, orange = foraging)
forage <- subset(enviro, ACT == 2)
bask <- subset(enviro, ACT == 1)
night <- subset(enviro, Solar < 1)
day <- subset(enviro, Solar >= 1)
with(night, plot(TIME ~ dates, xlim = xlim, ylab = "Hour of Day",
  xlab = "Day of Year", pch = 15, cex = 1, col = "dark blue"))

```

```
# nighttime hours
with(bask, points(TIME ~ dates, pch = 15, cex = 1, col = "light blue")) # basking T_bs
with(forage, points(TIME ~ dates, pch = 15, cex = 1, col = "orange")) # foraging T_bs
abline(v = as.POSIXct("2020-06-10"), col = "red", lty = 2, lwd = 2) # date of the workshop
```



References

- Kearney, M. R., and Porter, W. P. (2017). NicheMapR - an R package for biophysical modelling: the microclimate model. *Ecography*. doi:10.1111/ecog.02360
- Kearney, M. R., Gillingham, P. K., Bramer, I., Duffy, J. P., & Maclean, I. M. D. (2020). A method for computing hourly, historical, terrain-corrected microclimate anywhere on earth. *Methods in Ecology and Evolution*, 11(1), 38–43. <https://doi.org/10.1111/2041-210X.13330>
- Kemp M. U., Emiel van Loon E., Shamoun-Baranes J. & Bouten W. (2012) RNCEP: global weather and climate data at your fingertips: RNCEP. *Methods in Ecology and Evolution* 3 , 65–70.
- Klinges, D. H., Duffy, J. P., Kearney, M. R., & Maclean, I. M. D. (2022). mcera5: Driving microclimate models with ERA5 global gridded climate data. *Methods in Ecology and Evolution*, 13(n/a), 7. <https://doi.org/10.1111/2041-210X.13877>
- Maclean I. M. D., Mosedale J. R. & Bennie J. J. (2018) Microclima: an R package for modelling meso- and microclimate. *Methods in Ecology and Evolution* doi: 10.1111/2041-210X.13093.
- New M., Lister D., Hulme M. & Makin I. (2002) A high-resolution data set of surface climate over global land areas. *Climate Research* 21 , 1–25.