

heatcalc.R

Glenn Tattersall

Thu Jun 30 17:54:44 2016

```
# Estimating heat exchange from thermal images ####
# The purpose of this file is to provide a brief introduction to using Thermimage
# It is assumed that the user has thermal imaging experience and has extracted
# data from thermal images already, or has a means to bring thermal image data
# into R. This package will not replace thermal image analysis software but it
# can help in automating some calculations associated with large datasets of thermal
# images.

# Getting Started: Install Thermimage ####
# if you don't have Thermimage installed, type: install.packages("Thermimage")
# which should download Thermimage from the CRAN repository and install it.
# then simply type library(Thermimage) to call the functions into the working
# environment

library(Thermimage)

# Minimum required information (units) ####
# Surface temperatures, Ts (oC - note: all temperature units are in oC)
# Ambient temperatures, Ta (oC)
# Characteristic dimension of the object or animal, L (m)
# Surface Area, A (m^2)
# Shape of object: choose from "sphere", "hcylinder", "vcylinder", "hplate", "vplate"

# Required if working outdoors with solar radiation ####
# Visible surface reflectance, rho, which could be measured or estimated (0-1)
# Solar radiation (SE=abbrev for solar energy), W/m2

# Can be estimated or provided ####
# Wind speed, V (m/s) - I tend to model heat exchange under different V (0.1 to 10 m/s)
# Ground Temperature, Tg (oC) - estimated from air temperature if not provided
# Incoming infrared radiation, Ld (will be estimated from Air Temperature)
# Incoming infrared radiation, Lu (will be estimated from Ground Temperature)

# Ground Temperature Estimation ####
# For Ground Temp, we derived a relationship based on data in Galapagos that describes
# Tg-Ta ~ Se, (N=516, based on daytime measurements)
# Thus, Tg = 0.0187128*SE + Ta
# Derived from daytime measurements within the ranges:
# Range of Ta: 23.7 to 34 C
# Range of SE: 6.5 to 1506.0 Watts/m2
# Or from published work by Bartlett et al. (2006) in the Tground() function, the
# relationship would be Tg = 0.0121*SE + Ta

# Make your Data frame ####
# Once you have decided on what variables you have or need to model, create a data
# frame with these values (Ta, Ts, Tg, SE, A, L, Shape, rho), where each row corresponds to
# an individual measurement. The data frame is not required for calling functions,
```

```
# but it will force you to assemble your data before proceeding with calculations.
# Other records such as size, date image captured, time of day, species, sex, etc...
# should also be stored in the data frame.
```

```
Ta<-rnorm(20, 25, sd=10)
Ts<-Ta+rnorm(20, 5, sd=1)
RH<-rep(0.5, length(Ta))
SE<-rnorm(20, 400, sd=50)
Tg<-Tground(Ta,SE)
A<-rep(0.4,length(Ta))
L<-rep(0.1, length(Ta))
V<-rep(1, length(Ta))
shape<-rep("hcyllinder", length(Ta))
c<-forcedparameters(V=V, L=L, Ta=Ta, shape=shape)$c
n<-forcedparameters(V=V, L=L, Ta=Ta, shape=shape)$n
a<-freeparameters(L=L, Ts=Ts, Ta=Ta, shape=shape)$a
b<-freeparameters(L=L, Ts=Ts, Ta=Ta, shape=shape)$b
m<-freeparameters(L=L, Ts=Ts, Ta=Ta, shape=shape)$m
type<-rep("forced", length(Ta))
rho<-rep(0.1, length(Ta))
cloud<-rep(0, length(Ta))

d<-data.frame(Ta, Ts, Tg, SE, RH, rho, cloud, A, V, L, c, n, a, b, m, type, shape)
head(d)
```

```
##           Ta           Ts           Tg           SE  RH rho cloud  A V  L    c    n
## 1 13.20577 17.42638 18.72969 456.5220 0.5 0.1    0 0.4 1 0.1 0.174 0.618
## 2 36.30527 41.11712 41.82190 455.9203 0.5 0.1    0 0.4 1 0.1 0.174 0.618
## 3 23.34766 26.94144 27.68162 358.1789 0.5 0.1    0 0.4 1 0.1 0.174 0.618
## 4 29.15236 35.53570 34.63421 453.0451 0.5 0.1    0 0.4 1 0.1 0.174 0.618
## 5 25.59750 31.41907 30.64027 416.7577 0.5 0.1    0 0.4 1 0.1 0.174 0.618
## 6 16.65223 21.73243 20.45441 314.2294 0.5 0.1    0 0.4 1 0.1 0.174 0.618
##  a    b    m  type    shape
## 1 1 0.58 0.25 forced hcyllinder
## 2 1 0.58 0.25 forced hcyllinder
## 3 1 0.58 0.25 forced hcyllinder
## 4 1 0.58 0.25 forced hcyllinder
## 5 1 0.58 0.25 forced hcyllinder
## 6 1 0.58 0.25 forced hcyllinder
```

```
# Basic calculations #####
# The basic approach to estimating heat loss is based on that outlined in
# Tattersall et al (2009)
# This involves breaking the object into component shapes, deriving the exposed areas
# of those shapes empirically, and calculating Qtotal for each shape:
```

```
(Qtotal<-qgrad() + qconv())
```

```
## [1] -186.8849
```

```
# Notice how the above example yielded an estimate. This is because there are default
# values in all the functions. In this case, the estimate is negative, meaning
```

```
# a net loss of heat to the environment. It's units are in W/m2.  
# To convert the above measures into total heat flux, the Area of each part is required
```

```
Area1<-0.2 # units are in m2  
Area2<-0.3 # units are in m2  
Qtotal1<-qgrad()*Area1 + qconv()*Area1  
Qtotal2<-qgrad()*Area2 + qconv()*Area2  
QtotalAll<-Qtotal1 + Qtotal2
```

```
# This approach is used in animal thermal images, such that all component shapes sum to estimate entire  
# body heat exchange: WholeBody = Qtotal1 + Qtotal2 + Qtotal3 ... Qtotaln
```

```
# Qtotal is made up of two components: qgrad + qconv  
# qgrad is the net radiative heat flux (W/m2)  
# qconv is the net convective heat flux (W/m2)  
# qcond is usually ignored unless large contact areas between substrate. Additional  
# information is required to accurately calculate conductive heat exchange and are not  
# provided here.
```

```
# What is qabs #####  
# Radiation is both absorbed and emitted by animals. I have broken this down into  
# partially separate functions. qabs() is a function to estimate the area specific  
# amount of solar and infrared radiation absorbed by the object from the environment:  
# qabs() requires information on the air (ambient) temperature, ground temperature,  
# relative humidity, emissivity of the object, reflectivity of the object,  
# proportion cloud cover, and solar energy.  
qabs(Ta = 20, Tg = NULL, RH = 0.5, E = 0.96, rho = 0.1, cloud = 0, SE = 400)
```

```
## [1] 720.2545
```

```
# compare to a shaded environment with lower SE:  
qabs(Ta = 20, Tg = NULL, RH = 0.5, E = 0.96, rho = 0.1, cloud = 0, SE = 100)
```

```
## [1] 440.2954
```

```
# What is qgrad #####  
# Since the animal also emits radiation, qgrad() provides the net radiative heat  
# transfer.  
qgrad(Ts = 27, Ta = 20, Tg = NULL, RH = 0.5, E = 0.96, rho = 0.1, cloud = 0, SE = 100)
```

```
## [1] -1.486309
```

```
# Notice how the absorbed environmental radiation is ~440 W/m2, but the animal is also losing  
# losing a similar amount, so once we account for the net radiative flux, it very nearly  
# balances out at a slightly negative number (-1.486 W/m2)
```

```
# Ground temperature #####  
# If you have measured ground temperature, then simply include it in the call to qgrad:  
qgrad(Ts = 30, Ta = 25, Tg = 28, RH = 0.5, E = 0.96, rho = 0.1, cloud = 0, SE = 100)
```

```
## [1] 14.29534
```

```
# If you do not have ground temperature, but have measured Ta and SE, then let
# Tg=NULL. This will force a call to the Tground() function to estimate Tground
# It is likely better to assume that Tground is slightly higher than Ta, at least
# in the daytime. If using measurements obtained at night (SE=0), then you will have
# to provide both Ta and Tground, since Tground could be colder equal to Ta depending
# on cloud cover.
```

```
# What is hconv #####
# This is simply the convective heat coefficient. This is used in calculating the
# convective heat transfer and/or operative temperature but usually you will not need
# to call hconv() yourself
```

```
# What is qconv #####
# This is the function to calculate area specific convective heat transfer, analagous
# to grad, except for convective heat transfer. Positive values mean heat is gained
# by convection, negative values mean heat is lost by convection. Included in the
# function is the ability to estimate free convection (which occurs at 0 wind speed)
# or forced convection (wind speed >=0.1 m/s). Unless working in a completely still
# environment, it is more appropriate to used "forced" convection down to 0.1 m/s
# wind speed. Typical wind speeds indoors are likely <0.5 m/s, but outside can
# vary wildly.
# In addition to needing surface temperature, air temperature, and velocity, you
# need information/estimates on shape. L is the critical dimension of the shape,
# which is usually the height of an object within the air stream. The diameter of
# a horizontal cylinder is its critical dimension. Finally, shape needs to be
# assigned. see help(qconv) for details.
```

```
qconv(Ts = 30, Ta = 20, V = 1, L = 0.1, type = "forced", shape="hcylinder")
```

```
## [1] -102.6565
```

```
qconv(Ts = 30, Ta = 20, V = 1, L = 0.1, type = "forced", shape="hplate")
```

```
## [1] -124.323
```

```
qconv(Ts = 30, Ta = 20, V = 1, L = 0.1, type = "forced", shape="sphere")
```

```
## [1] -186.3256
```

```
# notice how the horizontal cylinder loses less than the horizontal plate which loses
# less than the sphere. Spherical objects lose ~1.8 times as much heat per area as
# cylinders.
```

```
# Which is higher: convection or radiation? #####
# Take a convection estimate at low wind speed:
```

```
qconv(Ts = 30, Ta = 20, V = 0.1, L = 0.1, type = "forced", shape="hcylinder")
```

```
## [1] -32.58495
```

```
# compare to a radiative estimate (without any solar absorption):
grad(Ts = 30, Ta = 20, Tg = NULL, RH = 0.5, E = 0.96, rho = 0.1, cloud = 0, SE = 0)
```

```
## [1] -112.6542
```

```
# in this case, the net radiative heat loss is greater than convective heat loss
# if you decrease the critical dimension, however, the convective heat loss per m2
# is much greater. This is effectively how convective exchange works: small objects
# lose heat from convection more readily than large objects (e.g. frostbite on fingers)
# If L is 10 times smaller:
qconv(Ts = 30, Ta = 20, V = 0.1, L = 0.01, type = "forced", shape="hcylinder")
```

```
## [1] -111.4338
```

```
grad(Ts = 30, Ta = 20, Tg = NULL, RH = 0.5, E = 0.96, rho = 0.1, cloud = 0, SE = 0)
```

```
## [1] -112.6542
```

```
# convection and radiative heat transfer are nearly the same.
# A safe conclusion here is that larger animals would rely more on radiative heat transfer
# than they would on convective heat transfer
```

```
# Sample Calculations #####
```

```
# Ideally, you have all parameters estimated or measured and put into a data frame.
```

```
# Using the dataframe, d we constructed earlier
```

```
(grad.A<-with(d, grad(Ts, Ta, Tg, RH, E=0.96, rho, cloud, SE)))
```

```
## [1] 348.0932 343.6387 259.0099 330.4094 300.2630 210.1902 273.0300
```

```
## [8] 379.5530 284.1577 242.0509 314.0573 293.1092 330.1466 239.2804
```

```
## [15] 306.0287 298.8722 328.5824 276.7348 266.3380 264.4860
```

```
(qconv.free.A<-with(d, qconv(Ts, Ta, V, L, c, n, a, b, m, type="free", shape)))
```

```
## [1] -16.55539 -19.30316 -13.46922 -27.55138 -24.59050 -20.83114 -20.77419
```

```
## [8] -31.98622 -20.82509 -20.23979 -16.23164 -22.03739 -18.26326 -22.38024
```

```
## [15] -19.12221 -22.34143 -19.94873 -23.78010 -24.27612 -22.50338
```

```
(qconv.forced.A<-with(d, qconv(Ts, Ta, V, L, c, n, a, b, m, type, shape)))
```

```
## [1] -43.71000 -48.52276 -36.74455 -64.84209 -59.36828 -52.37218 -51.76822
```

```
## [8] -73.52979 -51.73955 -50.51878 -42.74907 -54.06210 -47.12042 -55.49947
```

```
## [15] -48.62071 -54.66908 -49.89187 -57.89402 -59.08971 -55.00922
```

```
qtotal<-A*(grad.A + qconv.forced.A)
```

```
d<-data.frame(d, grad=grad.A*A, qconv=qconv.forced.A*A, qtotal=qtotal)
```

```
head(d)
```

```

##           Ta           Ts           Tg           SE RH rho cloud   A V   L       c       n
## 1 13.20577 17.42638 18.72969 456.5220 0.5 0.1   0 0.4 1 0.1 0.174 0.618
## 2 36.30527 41.11712 41.82190 455.9203 0.5 0.1   0 0.4 1 0.1 0.174 0.618
## 3 23.34766 26.94144 27.68162 358.1789 0.5 0.1   0 0.4 1 0.1 0.174 0.618
## 4 29.15236 35.53570 34.63421 453.0451 0.5 0.1   0 0.4 1 0.1 0.174 0.618
## 5 25.59750 31.41907 30.64027 416.7577 0.5 0.1   0 0.4 1 0.1 0.174 0.618
## 6 16.65223 21.73243 20.45441 314.2294 0.5 0.1   0 0.4 1 0.1 0.174 0.618
##   a   b   m   type   shape   grad   qconv   qtotal
## 1 1 0.58 0.25 forced hcyllinder 139.2373 -17.48400 121.75327
## 2 1 0.58 0.25 forced hcyllinder 137.4555 -19.40910 118.04639
## 3 1 0.58 0.25 forced hcyllinder 103.6040 -14.69782  88.90614
## 4 1 0.58 0.25 forced hcyllinder 132.1638 -25.93684 106.22694
## 5 1 0.58 0.25 forced hcyllinder 120.1052 -23.74731  96.35788
## 6 1 0.58 0.25 forced hcyllinder  84.0761 -20.94887  63.12722

```

```

# Test the equations out for consistency ####
# Toucan Proximal Bill at 10oC (from Tattersall et al 2009 spreadsheets)
A<-0.0097169
L<-0.0587
Ta<-10
Tg<-Ta
Ts<-15.59
SE<-0
rho<-0.1
E<-0.96
RH<-0.5
cloud<-1
V<-5
type="forced"
shape="hcyllinder"
(grad.A<-grad(Ts=Ts, Ta=Ta, Tg=Tg, RH=RH, E=E, rho=rho, cloud=cloud, SE=SE))

```

```
## [1] -37.90549
```

```

# compare to calculated value of -28.7 W/m2
# the R calculations differ slightly from Tattersall et al (2009) since they did not use
# estimates of longwave radiation (Ld and Lu), but instead assumed a simpler, constant Ta
# environment.
(grad.A<-grad(Ts=Ts, Ta=Ta, Tg=Tg, RH=RH, E=E, rho=rho, cloud=0, SE=SE))

```

```
## [1] -83.03191
```

```

# but if cloud = 0, then the grad values calculated here are much higher than calculated by
# Tattersall et al (2009) since they only estimated under simplifying, indoor conditions
# where background temperature = air temperature. In the outdoors, then, cloud presence
# would affect estimates of radiative heat loss
(qconv.forced.A<-qconv(Ts, Ta, V, L, type=type, shape=shape))

```

```
## [1] -192.7086
```

```
# compare to calculated value of -191.67 W/m2 - which is really close! The difference lies
# in estimates of air kinematic viscosity used
(qtotal.A<-(qrad.A + qconv.forced.A))
```

```
## [1] -275.7405
```

```
# Total area specific heat loss for the proximal area of the bill (Watts/m2)
qtotal.A*A
```

```
## [1] -2.679343
```

```
# Total heat loss for the proximal area of the bill (Watts)
# This lines up well with the published values in Tattersall et al (2009).
# This was confirmed in van de Van (2016) where they recalculated the area specific
# heat flux from toucan bills to be ~65 W/m2:
qrad(Ts=Ts, Ta=Ta, Tg=Tg, RH=0.5, E=0.96, rho=rho, cloud=1, SE=0) +
  qconv(Ts, Ta, V, L, type="free", shape=shape)
```

```
## [1] -64.83292
```

```
# Estimating Operative Temperature #####
# Operative environmental temperature is the expression of the "effective temperature" an
# object is experiencing, accounting for heat absorbed from radiation and heat lost to convection
# In other words, it is often used by some when trying to predict
# animal body temperature as a null expectation or reference point to determine whether
# active thermoregulation is being used. More often used in ectotherm studies, but as an
# initial estimate of what a freely moving animal temperature would be, it serves a useful
# reference. Usually, people would measure operative temperature with a model of an object
# placed into the environment, allowing wind, solar radiation and ambient temperature to
# influence its temperature. There are numerous formulations for it. The one here is from
# in Angilletta's book on Thermal Adaptations. Note: in the absence of sun or wind,
# operative temperature is simply ambient temperature.
```

```
# Operative temperature with varying reflectances:
```

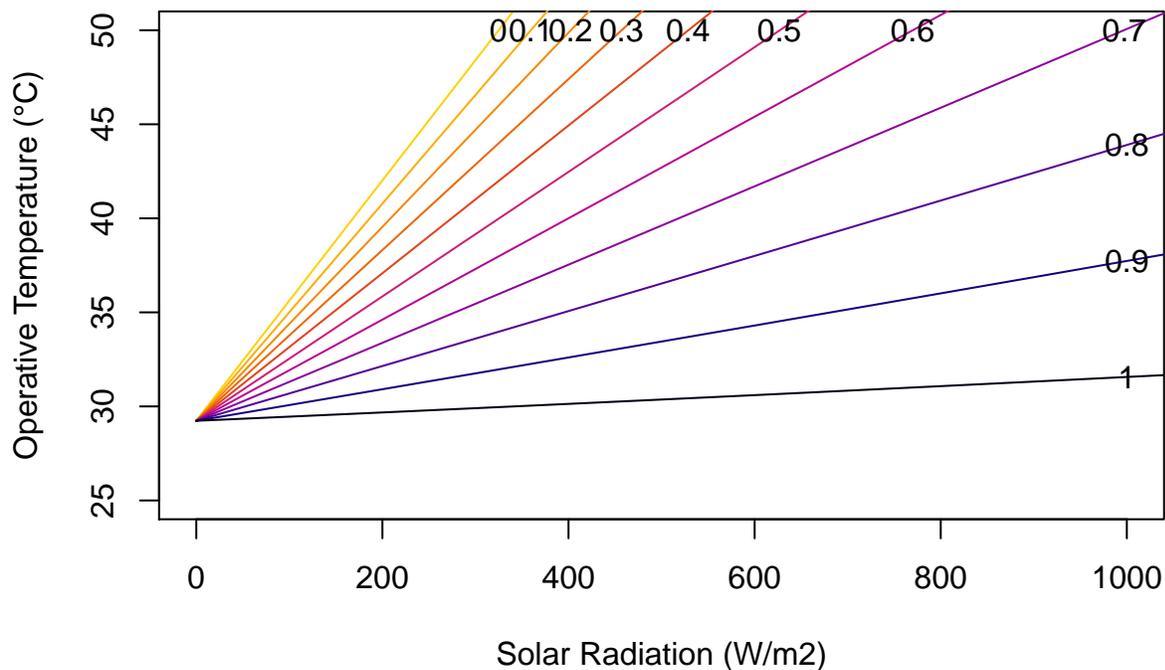
```
Ts<-40
Ta<-30
SE<-seq(0,1100,100)
Toperative<-NULL
for(rho in seq(0, 1, 0.1)){
  temp<-Te(Ts=Ts, Ta=Ta, Tg=NULL, RH=0.5, E=0.96, rho=rho, cloud=1, SE=SE, V=1,
    L=0.1, type="forced", shape="hcylinder")
  Toperative<-cbind(Toperative, temp)
}
rho<-seq(0, 1, 0.1)
Toperative<-data.frame(SE=seq(0,1100,100), Toperative)
colnames(Toperative)<-c("SE", seq(0,1,0.1))
matplot(Toperative$SE, Toperative[,-1], ylim=c(25, 50), type="l", xlim=c(0,1000),
  main="Effects of Altering Reflectance from 0 to 1",
  ylab="Operative Temperature (°C)", xlab="Solar Radiation (W/m2)", lty=1,
  col=flirpal[rev(seq(1,380,35))])
for(i in 2:12){
```

```

ymax<-par()$yaxp[2]
xmax<-par()$xaxp[2]
x<-Toperative[,1]; y<-Toperative[,i]
lm1<-lm(y~x)
b<-coefficients(lm1)[1]; m<-coefficients(lm1)[2]
if(max(y)>ymax) {xpos<-(ymax-b)/m; ypos<-ymax}
if(max(y)<ymax) {xpos<-xmax; ypos<-y[which(x==1000)]}
text(xpos, ypos, labels=rho[(i-1)])
}

```

Effects of Altering Reflectance from 0 to 1



```

# Operative temperature with varying wind speeds
Ts<-40
Ta<-30
SE<-seq(0,1100,100)
Toperative<-NULL
for(V in seq(0, 10, 1)){
  temp<-Te(Ts=Ts, Ta=Ta, Tg=NULL, RH=0.5, E=0.96, rho=0.1, cloud=1, SE=SE, V=V,
    L=0.1, type="forced", shape="vcylinder")
  Toperative<-cbind(Toperative, temp)
}
V<-seq(0, 10, 1)
Toperative<-data.frame(SE=seq(0,1100,100), Toperative)
colnames(Toperative)<-c("SE", seq(0,10,1))
matplot(Toperative$SE, Toperative[,-1], ylim=c(30, 50), type="l", xlim=c(0,1000),
  main="Effects of Altering Wind Speed from 0 to 10 m/s",
  ylab="Operative Temperature (°C)", xlab="Solar Radiation (W/m2)", lty=1,
  col=flirpal[rev(seq(1,380,35))])
for(i in 2:12){
  ymax<-par()$yaxp[2]

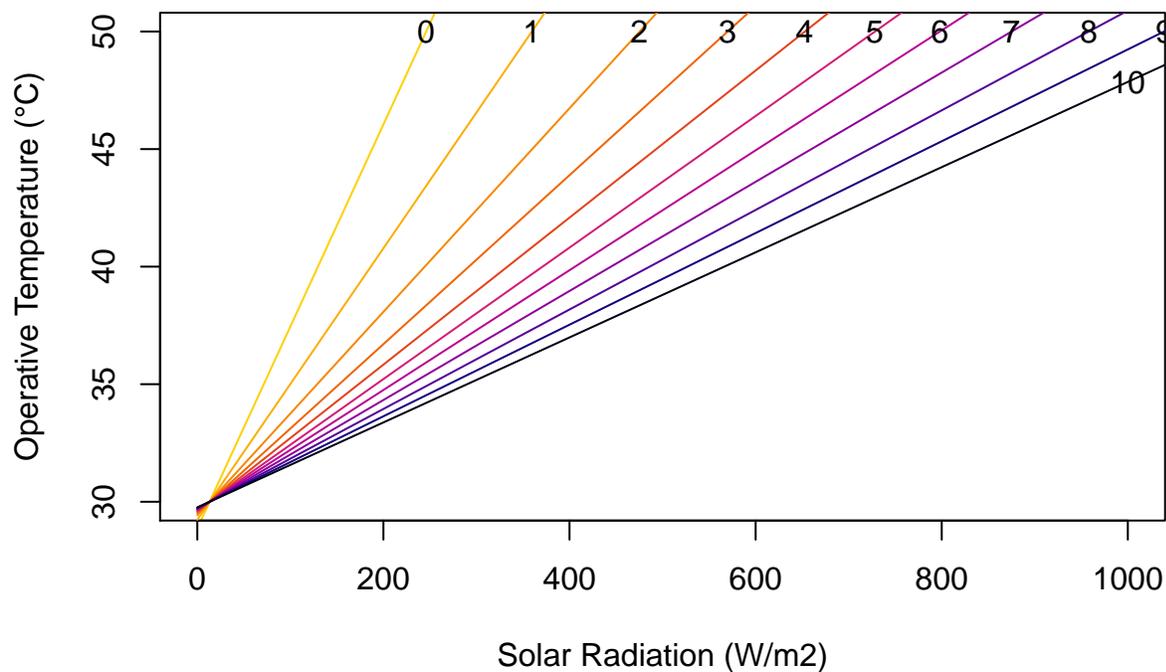
```

```

xmax<-par()$xaxp[2]
x<-Toperative[,1]; y<-Toperative[,i]
lm1<-lm(y~x)
b<-coefficients(lm1)[1]; m<-coefficients(lm1)[2]
if(max(y)>ymax) {xpos<-(ymax-b)/m; ypos<-ymax}
if(max(y)<ymax) {xpos<-xmax; ypos<-y[which(x==1000)]}
text(xpos, ypos, labels=V[(i-1)])
}

```

Effects of Altering Wind Speed from 0 to 10 m/s



```

# Operative temperature with varying RH
Ts<-40
Ta<-30
SE<-seq(0,1100,100)
Toperative<-NULL
for(RH in seq(0, 1, 0.1)){
  temp<-Te(Ts=Ts, Ta=Ta, Tg=NULL, RH=RH, E=0.96, rho=0.1, cloud=0.5, SE=SE, V=1,
    L=0.1, type="forced", shape="hcyllinder")
  Toperative<-cbind(Toperative, temp)
}
RH<-seq(0, 1, 0.1)
Toperative<-data.frame(SE=seq(0,1100,100), Toperative)
colnames(Toperative)<-c("SE", seq(0,1,0.1))
matplot(Toperative$SE, Toperative[,-1], ylim=c(30, 50), type="l", xlim=c(0,1000),
  main="Effects of changing RH from 0 to 1",
  ylab="Operative Temperature (°C)", xlab="Solar Radiation (W/m2)", lty=1,
  col=flirpal[rev(seq(1,380,35))])
for(i in 2:12){
  ymax<-par()$yaxp[2]
  xmax<-par()$xaxp[2]

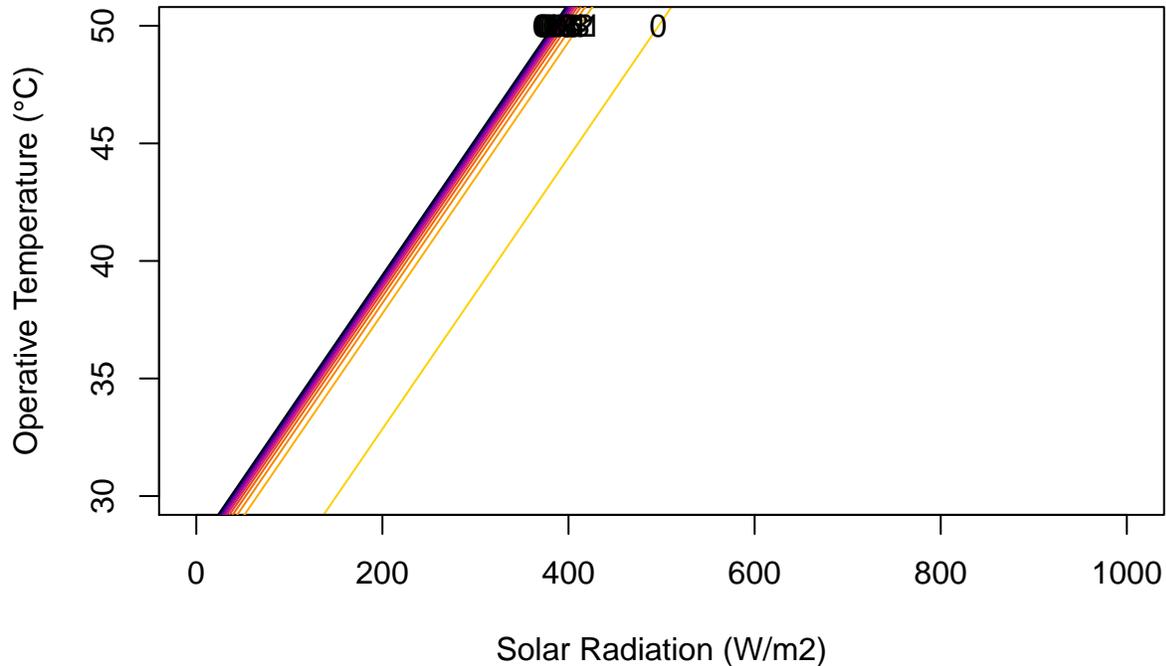
```

```

x<-Toperative[,1]; y<-Toperative[,i]
lm1<-lm(y~x)
b<-coefficients(lm1)[1]; m<-coefficients(lm1)[2]
if(max(y)>ymax) {xpos<-(ymax-b)/m; ypos<-ymax}
if(max(y)<ymin) {xpos<-xmax; ypos<-y[which(x==1000)]}
text(xpos, ypos, labels=RH[(i-1)])
}

```

Effects of changing RH from 0 to 1



```

# Operative temperature with varying cloud cover
Ts<-40
Ta<-30
SE<-seq(0,1100,100)
Toperative<-NULL
for(cloud in seq(0, 1, 0.1)){
  temp<-Te(Ts=Ts, Ta=Ta, Tg=NULL, RH=0.5, E=0.96, rho=0.5, cloud=cloud, SE=SE, V=1,
    L=0.1, type="forced", shape="hcylinder")
  Toperative<-cbind(Toperative, temp)
}
cloud<-seq(0, 1, 0.1)
Toperative<-data.frame(SE=seq(0,1100,100), Toperative)
colnames(Toperative)<-c("SE", seq(0,1,0.1))
matplot(Toperative$SE, Toperative[,-1], ylim=c(30, 50), type="l", xlim=c(0,1000),
  main="Effects of changing RH from 0 to 1",
  ylab="Operative Temperature (°C)", xlab="Solar Radiation (W/m2)", lty=1,
  col=flirpal[rev(seq(1,380,35))])
for(i in 2:12){
  ymax<-par()$yaxp[2]
  xmax<-par()$xaxp[2]
  x<-Toperative[,1]; y<-Toperative[,i]

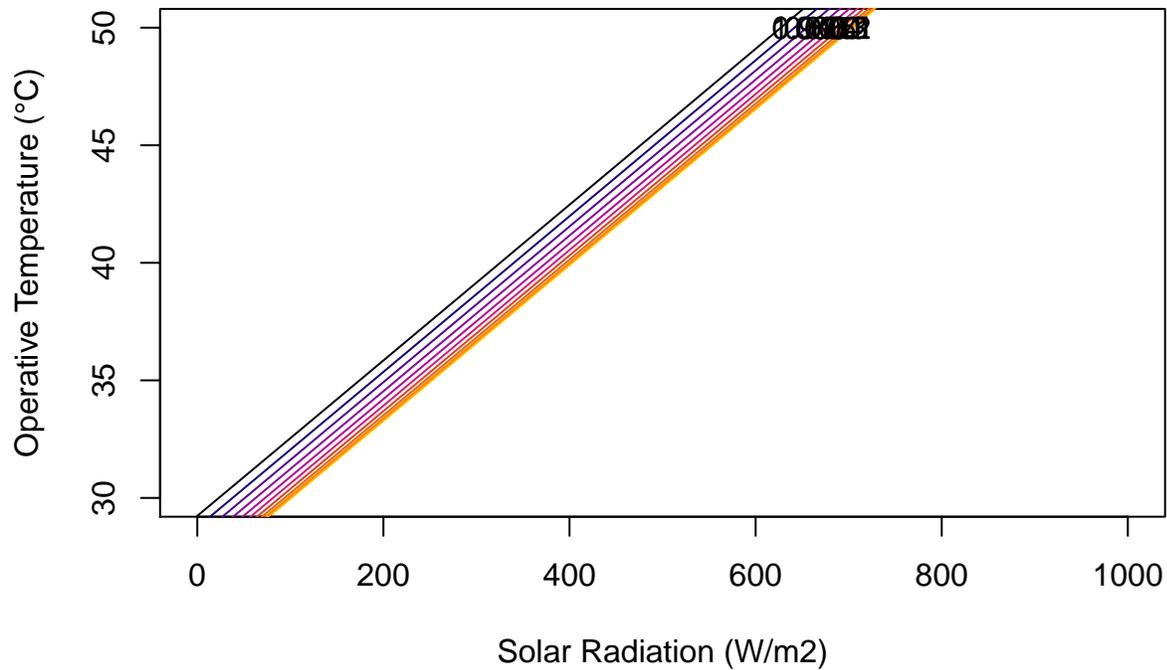
```

```

lm1<-lm(y~x)
b<-coefficients(lm1)[1]; m<-coefficients(lm1)[2]
if(max(y)>ymax) {xpos<-(ymax-b)/m; ypos<-ymax}
if(max(y)<ymin) {xpos<-xmax; ypos<-y[which(x==1000)]}
text(xpos, ypos, labels=cloud[(i-1)])
}

```

Effects of changing RH from 0 to 1



```

A<-0.0097169
L<-0.0587
Ta<-30
SE<-1000
Tg<-Tground(Ta, SE)
Ts<-41
E<-0.96
RH<-0.5
V<-1
type="forced"
shape="hcylinder"
(quad.A<-quad(Ts=Ts, Ta=Ta, Tg=Tg, RH=RH, E=E, rho=0.03, cloud=1, SE=SE))

```

```
## [1] 924.9294
```

```
(quad.A<-quad(Ts=Ts, Ta=Ta, Tg=Tg, RH=RH, E=E, rho=0.07, cloud=1, SE=SE))
```

```
## [1] 884.9294
```

```
(qconv.forced.A<-qconv(Ts, Ta, V, L, type=type, shape=shape))
```

```
## [1] -139.03
```

```
qconv(Ts, Ta, V, L, type="free", shape=shape)
```

```
## [1] -62.1259
```