moorea_heatwave

daniellembecker

1/15/2022

#OISST_data #Adapted by Amanda Chiachi #11/13/2019

Setup #load libraries

```
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
```

```
library(ggplot2)
library(heatwaveR)# The packages we will use
library(dplyr) # A staple for modern data management in R
library(lubridate) # Useful functions for dealing with dates
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2) # The preferred library for data visualisation
library(tidync) # For easily dealing with NetCDF data
library(rerddap) # For easily downloading subsets of data
library(doParallel) # For parallel processing
```

```
## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel
```

```
library(lubridate)
library(scales)
library(maps)
```

The information for the NOAA OISST data

```
rerddap::info(datasetid = "ncdcOisst21Agg_LonPM180", url = "https://coastwatch.pfeg.noaa.gov/erddap/")
```

```
## <ERDDAP info> ncdcOisst21Agg_LonPM180
##  Base URL: https://coastwatch.pfeg.noaa.gov/erddap
##  Dataset Type: griddap
##  Dimensions (range):
##      time: (1981-09-01T12:00:00Z, 2022-01-10T12:00:00Z)
##      zlev: (0.0, 0.0)
##      latitude: (-89.875, 89.875)
##      longitude: (-179.875, 179.875)
##  Variables:
##      anom:
##          Units: degree_C
##      err:
##          Units: degree_C
##      ice:
##          Units: 1
##      sst:
##          Units: degree_C
```

This function expects the user to provide it with a start and end date

It then downloads and prepares the data

we are downloading only the SST data out of several variables

spatial extent of latitude -17.466987, -17.597282 & longitude -149.849395, -149.822959

```
OISST_sub_dl <- function(time_df){
  OISST_dat <- griddap(x = "ncdcOisst21Agg_LonPM180",
                       url = "https://coastwatch.pfeg.noaa.gov/erddap/",
                       time = c(time_df$start, time_df$end),
                       zlev = c(0, 0),
                       latitude = c(-17.466987, -17.597282),
                       longitude = c(-149.849395, -149.822959),
                       fields = "sst")$data %>%
    mutate(time = as.Date(stringr::str_remove(time, "T00:00:00Z"))) %>%
    dplyr::rename(t = time, temp = sst) %>%
    select(lon, lat, t, temp) %>%
    na.omit()
}
```

#server doesnt like more than 9 years of consecutive data #creates a data frame to download multiple batches of the data # allows us to automate the entire download # Date download range by start and end dates per year

```
moorea_years <- data.frame(date_index = 1,
                                 start = as.Date(c("2002-01-01")),
                                 end = as.Date(c("2021-11-01")))



system.time(
  OISST_data <- moorea_years %>%
    group_by(date_index) %>%
    group_modify(~OISST_sub_dl(.x)) %>%
    ungroup() %>%
    select(lon, lat, t, temp)
)
```

```
##    user  system elapsed
##   2.249   0.093   3.860
```

```
#format date and time, lubridate package format
OISST_data$t <- ymd(OISST_data$t)

#subset temp data for March through April for 2019 and 2020

heatwave.temps <- OISST_data %>%
  filter(lubridate::month(t) %in% c(3:4))

heatwave.temps_filter <- heatwave.temps %>%
  filter(lubridate::year(t) %in% c(2019))

write.csv(heatwave.temps_filter, "RAnalysis/output/moorea.heatwave.temps.csv")
```

Detect the events in a time series

“. . . must be at least three years to calculate thresholds”

create your time series data

```
ts.OISST <- ts2clm(OISST_data, climatologyPeriod = c("2002-01-01", "2021-11-01"))
mhw.OISST <- detect_event(ts.OISST)
```

View just a few metrics

The function will return a line plot indicating the climatology, threshold and temperature, with the hot or cold events that meet the specifications of Hobday et al. (2016) shaded in as appropriate.

An anomalously warm event is considered to be a MHW if it lasts for five or more days, with temperatures warmer than the 90th percentile based on a 30-year historical baseline period.

```
#make data frame with marine heatwave events over time in Mo'orea

mhw.record.OISST <- mhw.OISST$event %>%
  dplyr::ungroup() %>%
  dplyr::select(event_no, duration, date_start, date_end, date_peak, intensity_max, intensity_cumulative
  dplyr::arrange(-intensity_max); mhw.record.OISST
```

```
## # A tibble: 87 x 10
##    event_no duration date_start date_end   date_peak  intensity_max
##       <int>    <dbl> <date>     <date>     <date>             <dbl>
##  1       43        5 2014-10-27 2014-10-29 2014-10-28          2.17
##  2       62       30 2017-01-05 2017-01-19 2017-01-09          1.53
##  3       75       15 2020-03-07 2020-03-14 2020-03-10          1.49
##  4       60       13 2016-11-05 2016-11-11 2016-11-10          1.48
##  5       45       18 2015-01-12 2015-01-20 2015-01-13          1.44
##  6       17       22 2005-12-31 2006-01-10 2006-01-06          1.43
##  7       10        6 2003-05-17 2003-05-19 2003-05-19          1.43
##  8       71       55 2019-04-01 2019-04-28 2019-04-22          1.40
##  9        7       12 2003-03-15 2003-03-20 2003-03-19          1.38
## 10       14        5 2005-11-27 2005-11-29 2005-11-27          1.38
## # ... with 77 more rows, and 4 more variables: intensity_cumulative <dbl>,
## #   intensity_mean <dbl>, rate_onset <dbl>, rate_decline <dbl>
```

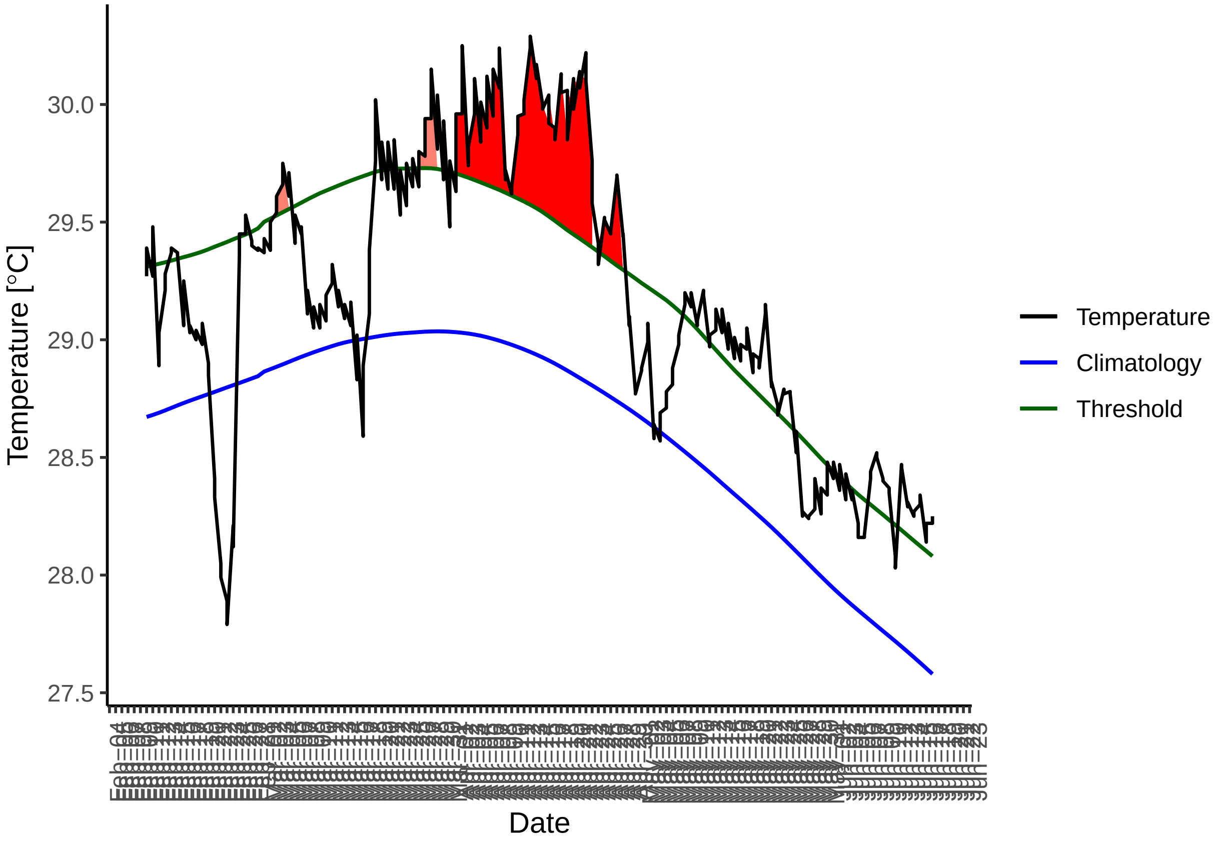
```
write.csv(mhw.record.OISST, "RAnalysis/output/marine.heatwave.record.OISST.csv")

#format date and time
mhw.OISST$Date<- mdy(mhw.OISST$Date)


#create a line plot of marine heatwave data, can change date range for more specific times
#2019 heatwave scenarios
event_line(mhw.OISST, spread = 50, metric = "intensity_max",
           start_date = "2019-03-01", end_date = "2019-05-01", x_axis_title = "Date") +
    scale_x_date(date_breaks = "1 day", labels = date_format("%b-%d")) +
    theme_classic()+
    theme(axis.text.x = element_text(angle = 90))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

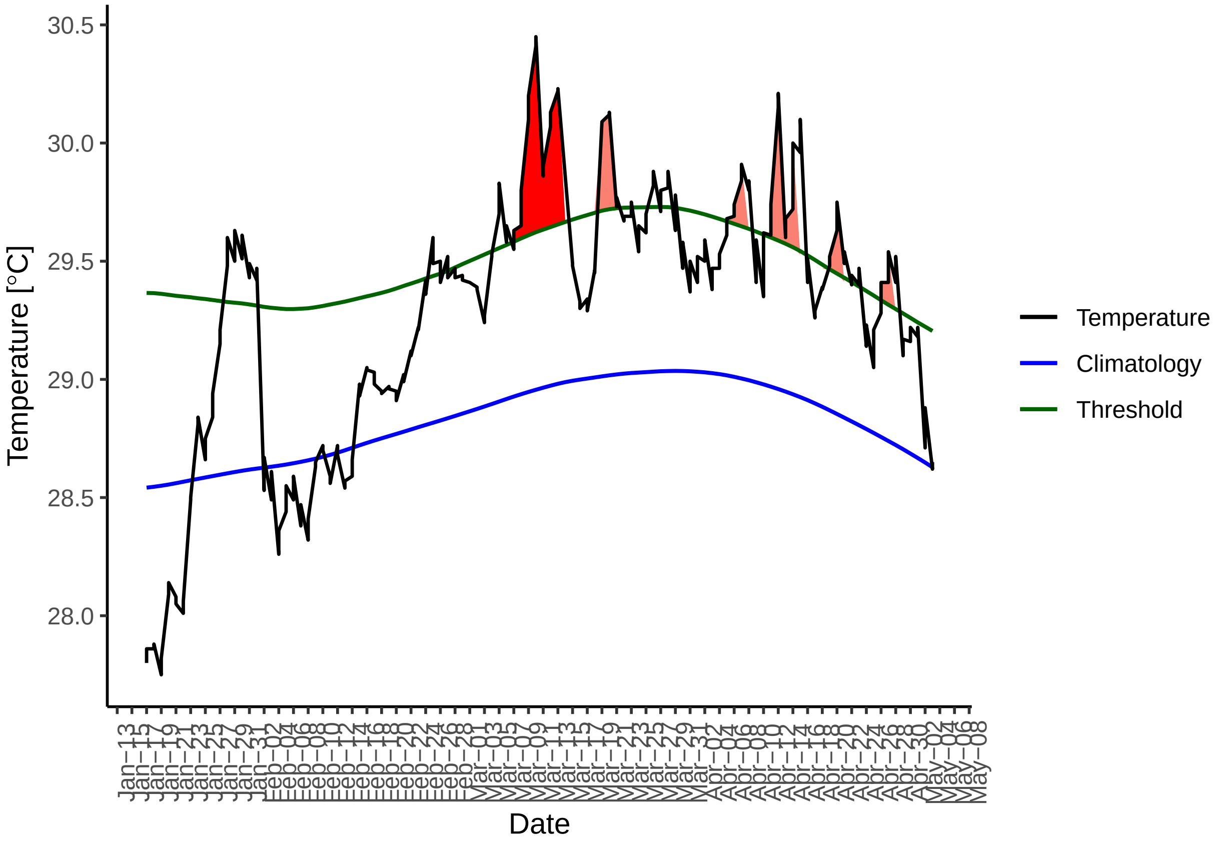
```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```



```
#2020 heatwave scenarios
event_line(mhw.OISST, spread = 50, metric = "intensity_max",
           start_date = "2020-03-01", end_date = "2020-05-01", x_axis_title = "Date") +
    scale_x_date(date_breaks = "2 days", labels = date_format("%b-%d")) +
    theme_classic()+
    theme(axis.text.x = element_text(angle = 90))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

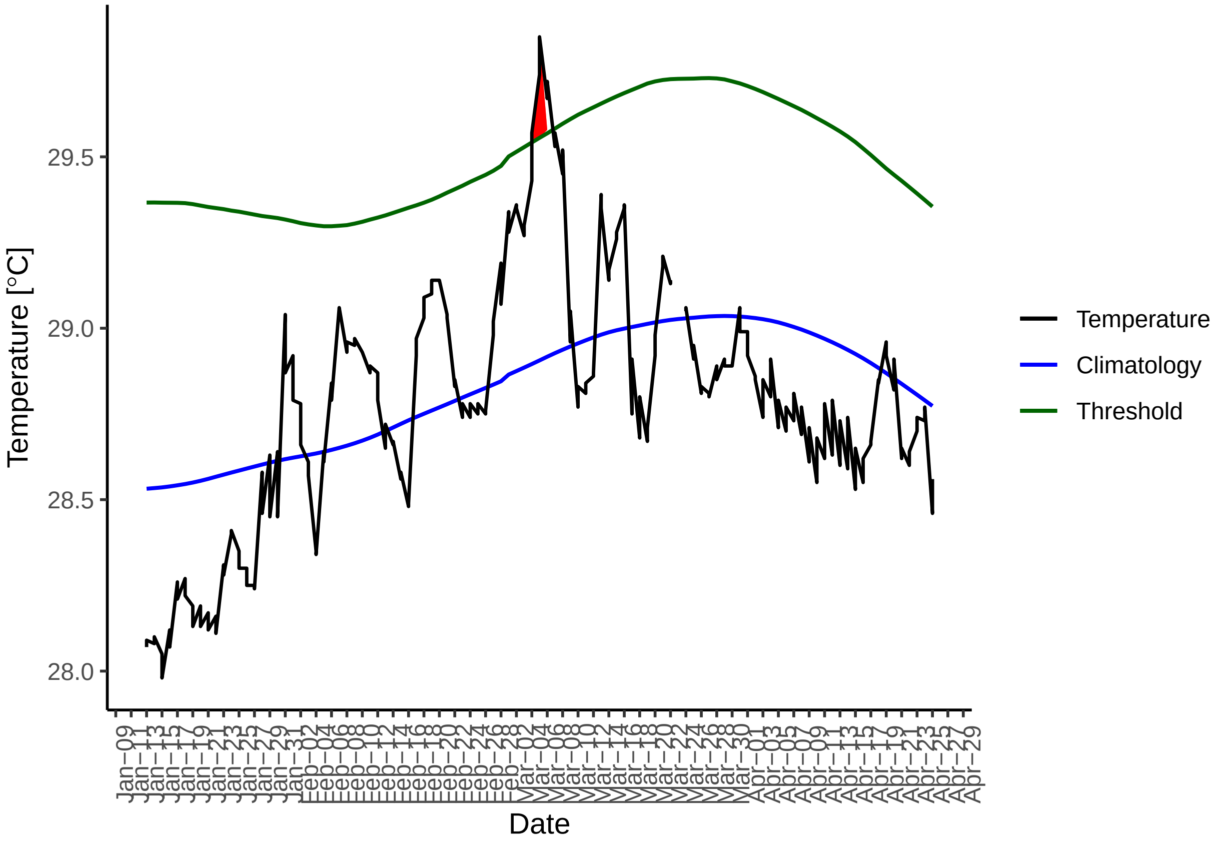
```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```



```
#2021 heatwave scenarios
event_line(mhw.OISST, spread = 50, metric = "intensity_max",
           start_date = "2021-03-01", end_date = "2021-05-01", x_axis_title = "Date") +
    scale_x_date(date_breaks = "2 days", labels = date_format("%b-%d")) +
    theme_classic()+
    theme(axis.text.x = element_text(angle = 90))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```



####Use heatwave package on LTER 02 backreef temperature data

#format LTER2 dataframe to use for mhw scenario

```
#load in LTER 02 backreef data from 2005 to 2021, delete first row
LTER2 <- read.csv("RAnalysis/output/LTER_2_temp_backreef.csv", row.names = 1)

#remove columns not needed for analysis (site,time,reef.type,and depth)

LTER2.mhw <- LTER2[ -c(1,3:5) ]

#rename date column to t

LTER2.mhw <- rename(LTER2.mhw, t = Date)

#need to get data point per day to match OISST processing

LTER2.mhw.dat <- LTER2.mhw %>%
  group_by(t) %>% #tells to group by treatment
  summarise(mean=mean(temp, na.rm=TRUE)) #calculates mean and se
LTER2.mhw.dat
```

```
## # A tibble: 5,589 x 2
##    t           mean
##    <chr>      <dbl>
##  1 2005-06-01  27.7
##  2 2005-06-02  27.7
##  3 2005-06-03  27.7
##  4 2005-06-04  27.6
##  5 2005-06-05  27.6
##  6 2005-06-06  27.6
##  7 2005-06-07  27.4
##  8 2005-06-08  27.1
##  9 2005-06-09  26.7
## 10 2005-06-10  26.4
## # ... with 5,579 more rows
```

```
#add lat and lon columns to data frame

LTER2.mhw <- LTER2.mhw.dat %>%
  mutate(lon = -149.8116849,)

LTER2.mhw <- LTER2.mhw %>%
  mutate(lat = -17.48131958)

#rearrange columns in order for mhw scenario

LTER2.mhw.reorder <- LTER2.mhw[, c(3,4, 1, 2)]

LTER2.mhw.reorder
```

```
## # A tibble: 5,589 x 4
##       lon   lat t           mean
##     <dbl> <dbl> <chr>      <dbl>
##  1 -150. -17.5 2005-06-01  27.7
##  2 -150. -17.5 2005-06-02  27.7
##  3 -150. -17.5 2005-06-03  27.7
##  4 -150. -17.5 2005-06-04  27.6
##  5 -150. -17.5 2005-06-05  27.6
##  6 -150. -17.5 2005-06-06  27.6
##  7 -150. -17.5 2005-06-07  27.4
##  8 -150. -17.5 2005-06-08  27.1
##  9 -150. -17.5 2005-06-09  26.7
## 10 -150. -17.5 2005-06-10  26.4
## # ... with 5,579 more rows
```

```
#rename mean column to temp

LTER2.mhw <- rename(LTER2.mhw.reorder, temp = mean)

#format date and time, lubridate package format
LTER2.mhw$t <- ymd(LTER2.mhw$t)

LTER2.mhw
```

```
## # A tibble: 5,589 x 4
##       lon   lat t           temp
##     <dbl> <dbl> <date>     <dbl>
##  1 -150. -17.5 2005-06-01  27.7
##  2 -150. -17.5 2005-06-02  27.7
##  3 -150. -17.5 2005-06-03  27.7
##  4 -150. -17.5 2005-06-04  27.6
##  5 -150. -17.5 2005-06-05  27.6
##  6 -150. -17.5 2005-06-06  27.6
##  7 -150. -17.5 2005-06-07  27.4
##  8 -150. -17.5 2005-06-08  27.1
##  9 -150. -17.5 2005-06-09  26.7
## 10 -150. -17.5 2005-06-10  26.4
## # ... with 5,579 more rows
```

Detect the events in a time series

create your time series data

```
ts.lter <- ts2clm(LTER2.mhw, climatologyPeriod = c("2005-06-01", "2021-07-06"))
mhw.lter <- detect_event(ts.lter)
```

View just a few metrics

The function will return a line plot indicating the climatology, threshold and temperature, with the hot or cold events that meet the specifications of Hobday et al. (2016) shaded in as appropriate.

An anomalously warm event is considered to be a MHW if it lasts for five or more days, with temperatures warmer than the 90th percentile based on a 30-year historical baseline period.

```
#make data frame with marine heatwave events over time in Mo'orea

mhw.record.lter <- mhw.lter$event %>%
  dplyr::ungroup() %>%
  dplyr::select(event_no, duration, date_start, date_end, date_peak, intensity_max, intensity_cumulative
  dplyr::arrange(-intensity_max); mhw.record.lter
```

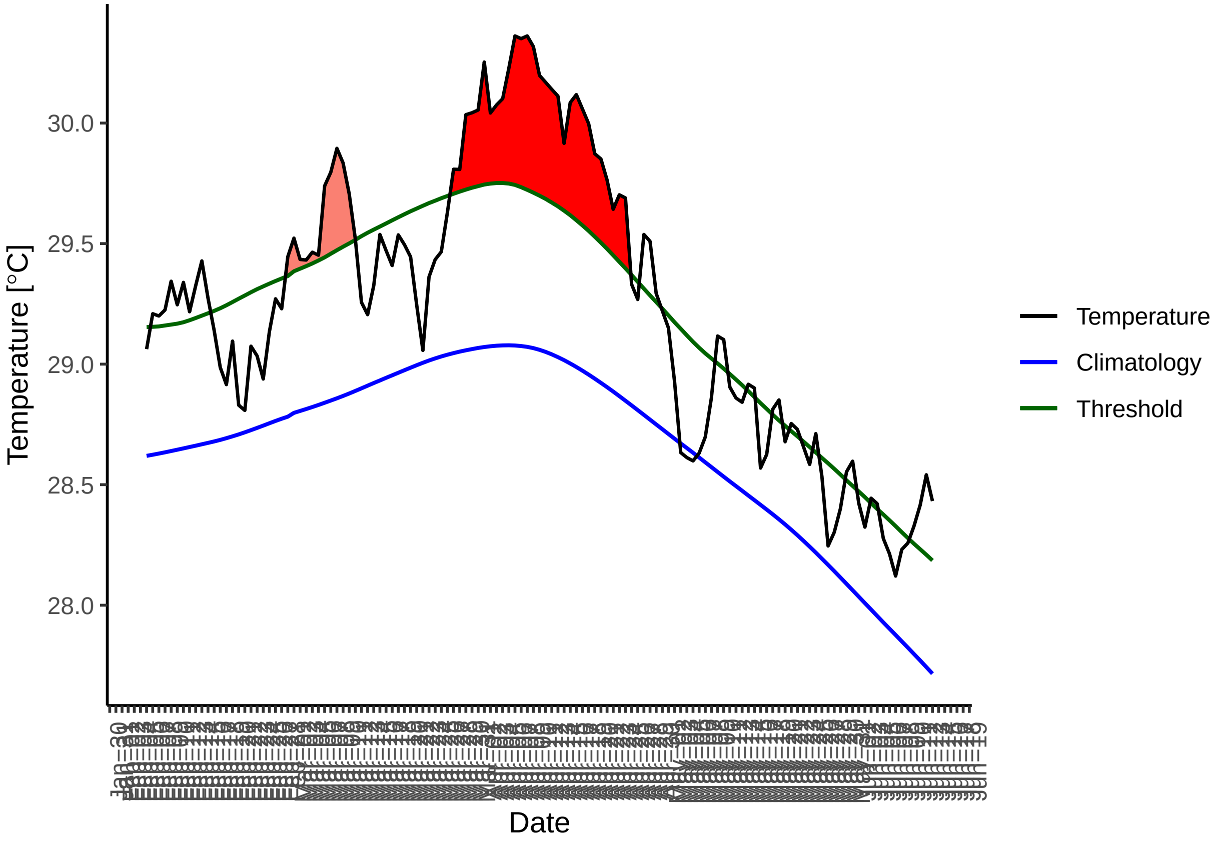
```
## # A tibble: 39 x 10
##    event_no duration date_start date_end   date_peak  intensity_max
##       <int>    <dbl> <date>     <date>     <date>             <dbl>
##  1        3       13 2005-11-25 2005-12-07 2005-11-29          1.38
##  2       27       13 2018-12-13 2018-12-25 2018-12-19          1.33
##  3       19       50 2016-09-23 2016-11-11 2016-09-26          1.31
##  4       32       29 2019-03-27 2019-04-24 2019-04-08          1.29
##  5       16        8 2015-01-10 2015-01-17 2015-01-13          1.24
##  6       14        6 2012-11-29 2012-12-04 2012-12-01          1.23
##  7       21        6 2017-01-05 2017-01-10 2017-01-09          1.21
##  8       24       33 2017-07-07 2017-08-08 2017-07-07          1.15
##  9       28        8 2018-12-31 2019-01-07 2019-01-03          1.15
## 10       26       12 2018-11-25 2018-12-06 2018-11-26          1.13
## # ... with 29 more rows, and 4 more variables: intensity_cumulative <dbl>,
## #   intensity_mean <dbl>, rate_onset <dbl>, rate_decline <dbl>
```

```
write.csv(mhw.record.lter, "RAnalysis/output/marine.heatwave.record.lter.csv")


#create a line plot of marine heatwave data, can change date range for more specific times
#2019 heatwave scenarios
event_line(mhw.lter, spread = 50, metric = "intensity_max",
           start_date = "2019-03-01", end_date = "2019-05-01", x_axis_title = "Date") +
    scale_x_date(date_breaks = "1 day", labels = date_format("%b-%d")) +
    theme_classic()+
    theme(axis.text.x = element_text(angle = 90))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

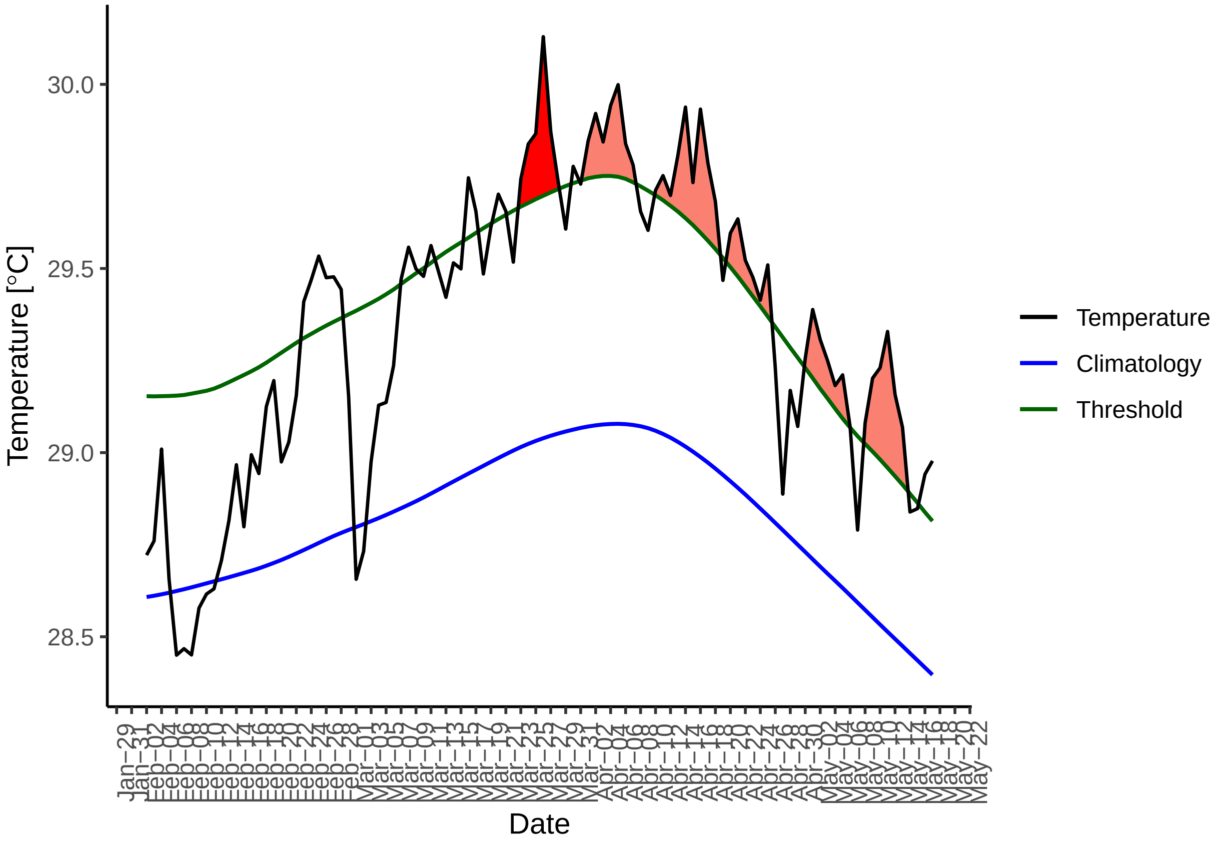
```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```



```
#2020 heatwave scenarios
event_line(mhw.lter, spread = 50, metric = "intensity_max",
           start_date = "2020-03-01", end_date = "2020-05-01", x_axis_title = "Date") +
    scale_x_date(date_breaks = "2 days", labels = date_format("%b-%d")) +
    theme_classic()+
    theme(axis.text.x = element_text(angle = 90))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```


Temperature [°C]
30.0
29.5
29.0
28.5
Jan−29
Jan−31
Feb−02
Feb−04
Feb−06
Feb−08
Feb−10
Feb−12
Feb−14
Feb−16
Feb−18
Feb−20
Feb−22
Feb−24
Feb−26
Feb−28
Mar−01
Mar−03
Mar−05
Mar−07
Mar−09
Mar−11
Mar−13
Mar−15
Mar−17
Mar−19
Mar−21
Mar−23
Mar−25
Mar−27
Mar−29
Mar−31
Apr−02
Apr−04
Apr−06
Apr−08
Apr−10
Apr−12
Apr−14
Apr−16
Apr−18
Apr−20
Apr−22
Apr−24
Apr−26
Apr−28
Apr−30
May−02
May−04
May−06
May−08
May−10
May−12
May−14
May−16
May−18
May−20
May−22
Date
Temperature
Climatology
Threshold