

Weather Data

JcB

28/07/2015

Contents

1	Librairie weatherData	1
2	Station météo	1
2.1	Alsace	1
2.2	Damgan	1
2.3	Aéroport de Vannes	2
3	Afficher la météo du jour	2
4	Température du mois	5
5	Quand déclare-t-on l'état de canicule ?	9

1 Librairie weatherData

source

initialisation:

```
library("weatherData")
```

Utilise: - le code s'une station météo sous la forme d'un sigle aéroportuaire: Strasbourg = LFST - la date du jour

2 Station météo

2.1 Alsace

- STRASBOURG/ENTZH **LFST**
- BALE/MULHOUSE **LFSB**
- COLMAR/MEYENHEIM **LFSC**

2.2 Damgan

- <http://www.meteo-damgan.fr/> météo complète + horaires des marées
- indicatif de la station: IBRETAGN30

2.3 Aéroport de Vannes

- C'est le plus proche de Pluherlin
- indicatif: LFRV

3 Afficher la météo du jour

La méthode `getDetailedWeather` donne les infos le jour donné sous forme d'un dataframe.

- `opt_all_columns = TRUE` affiche toutes les infos disponibles par heure. Par défaut (`FALSE`) n'affiche que les températures.
- `opt_custom_columns` permet un choix personnalisé des colonnes à afficher. Par défaut vaut `FALSE`. Si on met `TRUE`, il faut obligatoirement renseigner l'option `_custom_columns`.
- `custom_columns` vecteur d'`INTEGER` indiquant le n° de la colonne à afficher. La première colonne (*Time*) est toujours affichée. Il est donc inutile de le préciser. La numérotation effective commence à 2. Par exemple or afficher la température et la pression: `names(getDetailedWeather("LFST", "2014-06-10", opt_custom_columns=TRUE, custom_columns=c(2,5)))`

```
today <- format(Sys.time(), "%Y-%m-%d")
station <- "LFST"
```

```
checkDataAvailability(station_id = "LFST", check_date = today)
```

```
## Getting data from:
## http://www.wunderground.com/history/airport/LFST/2015/7/28/DailyHistory.html?format=1
##
## The following columns are available for:2015-07-28
```

```
## [1] "TimeCEST"           "TemperatureC"
## [3] "Dew_PointC"         "Humidity"
## [5] "Sea_Level_PressurehPa" "VisibilityKm"
## [7] "Wind_Direction"     "Wind_SpeedKm_h"
## [9] "Gust_SpeedKm_h"     "Precipitationmm"
## [11] "Events"             "Conditions"
## [13] "WindDirDegrees"     "DateUTC"
```

```
## Checking Data Availability For LFST
## Found Records for 2015-07-28
## Data is Available
```

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

```
station <- "LFST"# strasbourg
```

```
t <- getDetailedWeather(station, today, opt_all_columns = TRUE)
names(t)
```

```
## [1] "Time"               "TimeCEST"
## [3] "TemperatureC"       "Dew_PointC"
## [5] "Humidity"           "Sea_Level_PressurehPa"
```

```
## [7] "VisibilityKm"          "Wind_Direction"
## [9] "Wind_SpeedKm_h"       "Gust_SpeedKm_h"
## [11] "Precipitationmm"      "Events"
## [13] "Conditions"           "WindDirDegrees"
## [15] "DateUTC"
```

```
str(t)
```

```
## 'data.frame': 60 obs. of 15 variables:
## $ Time : POSIXct, format: "2015-07-28 00:00:00" "2015-07-28 00:00:00" ...
## $ TimeCEST : chr "12:00 AM" "12:00 AM" "12:30 AM" "1:00 AM" ...
## $ TemperatureC : num 19 19 20 20 20 21 20 21 20 20 ...
## $ Dew_PointC : num 10 10 10 11 11 11 11 11 11 11 ...
## $ Humidity : int 44 56 52 48 56 53 44 53 56 47 ...
## $ Sea_Level_PressurehPa: int 1009 1009 1009 1009 1009 1009 1009 1009 1009 1009 ...
## $ VisibilityKm : num 60 -9999 10 60 10 ...
## $ Wind_Direction : chr "SSW" "SSW" "SSW" "SSW" ...
## $ Wind_SpeedKm_h : num 13 13 13 11.1 11.1 24.1 18.5 18.5 16.7 14.8 ...
## $ Gust_SpeedKm_h : chr "" "-" "-" "" ...
## $ Precipitationmm : chr "" "N/A" "N/A" "" ...
## $ Events : logi NA NA NA NA NA NA ...
## $ Conditions : chr "Clear" "Clear" "Overcast" "Clear" ...
## $ WindDirDegrees : int 210 210 200 210 210 250 250 250 280 270 ...
## $ DateUTC : chr "2015-07-27 22:00:00" "2015-07-27 22:00:00" "2015-07-27 22:30:00" "2015-07-27 23:00:00"
```

```
head(t)
```

```
##           Time TimeCEST TemperatureC Dew_PointC Humidity
## 1 2015-07-28 00:00:00 12:00 AM          19         10      44
## 2 2015-07-28 00:00:00 12:00 AM          19         10      56
## 3 2015-07-28 00:30:00 12:30 AM          20         10      52
## 4 2015-07-28 01:00:00 1:00 AM          20         11      48
## 5 2015-07-28 01:00:00 1:00 AM          20         11      56
## 6 2015-07-28 01:30:00 1:30 AM          21         11      53
##   Sea_Level_PressurehPa VisibilityKm Wind_Direction Wind_SpeedKm_h
## 1              1009           60          SSW          13.0
## 2              1009          -9999          SSW          13.0
## 3              1009           10          SSW          13.0
## 4              1009           60          SSW          11.1
## 5              1009           10          SSW          11.1
## 6              1009           10          WSW          24.1
##   Gust_SpeedKm_h Precipitationmm Events Conditions WindDirDegrees
## 1              -              NA      Clear          210
## 2              -              N/A      Clear          210
## 3              -              N/A      Overcast        200
## 4              -              NA      Clear          210
## 5              -              N/A      Overcast        210
## 6              -              N/A      Overcast        250
##           DateUTC
## 1 2015-07-27 22:00:00
## 2 2015-07-27 22:00:00
## 3 2015-07-27 22:30:00
## 4 2015-07-27 23:00:00
```

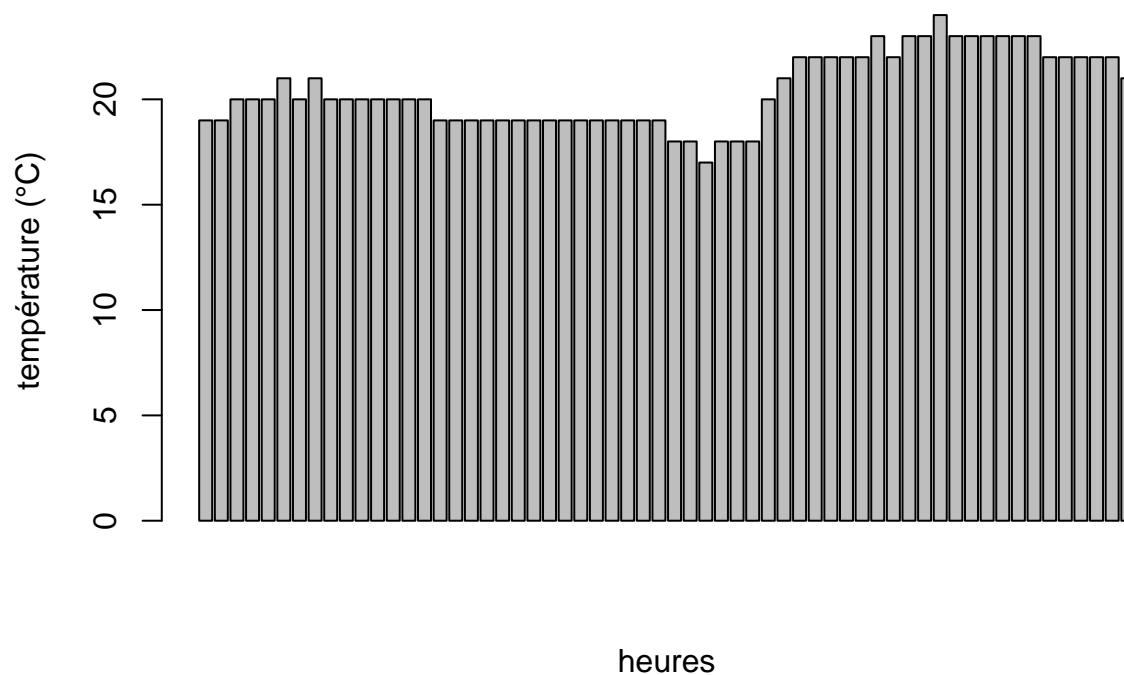
```
## 5 2015-07-27 23:00:00
## 6 2015-07-27 23:30:00
```

```
t[, c("TimeCEST", "TemperatureC", "Sea_Level_PressurehPa")]
```

##	TimeCEST	TemperatureC	Sea_Level_PressurehPa
## 1	12:00 AM	19	1009
## 2	12:00 AM	19	1009
## 3	12:30 AM	20	1009
## 4	1:00 AM	20	1009
## 5	1:00 AM	20	1009
## 6	1:30 AM	21	1009
## 7	2:00 AM	20	1009
## 8	2:00 AM	21	1009
## 9	2:30 AM	20	1009
## 10	3:00 AM	20	1009
## 11	3:00 AM	20	1009
## 12	3:30 AM	20	1009
## 13	4:00 AM	20	1009
## 14	4:00 AM	20	1009
## 15	4:30 AM	20	1009
## 16	5:00 AM	19	1009
## 17	5:00 AM	19	1009
## 18	5:30 AM	19	1009
## 19	6:00 AM	19	1010
## 20	6:00 AM	19	1010
## 21	6:30 AM	19	1010
## 22	7:00 AM	19	1010
## 23	7:00 AM	19	1010
## 24	7:30 AM	19	1010
## 25	8:00 AM	19	1011
## 26	8:00 AM	19	1010
## 27	8:30 AM	19	1011
## 28	9:00 AM	19	1011
## 29	9:00 AM	19	1011
## 30	9:30 AM	19	1011
## 31	10:00 AM	18	1012
## 32	10:00 AM	18	1011
## 33	10:30 AM	17	1011
## 34	11:00 AM	18	1012
## 35	11:00 AM	18	1011
## 36	11:30 AM	18	1011
## 37	12:00 PM	20	1011
## 38	12:00 PM	21	1011
## 39	12:30 PM	22	1011
## 40	1:00 PM	22	1011
## 41	1:00 PM	22	1011
## 42	1:30 PM	22	1011
## 43	2:00 PM	22	1010
## 44	2:00 PM	23	1010
## 45	2:30 PM	22	1010
## 46	3:00 PM	23	1010
## 47	3:00 PM	23	1010
## 48	3:30 PM	24	1010

```
## 49 4:00 PM      23      1010
## 50 4:00 PM      23      1010
## 51 4:30 PM      23      1010
## 52 5:00 PM      23      1010
## 53 5:00 PM      23      1010
## 54 5:30 PM      23      1010
## 55 6:00 PM      22      1010
## 56 6:00 PM      22      1010
## 57 6:30 PM      22      1010
## 58 7:00 PM      22      1010
## 59 7:00 PM      22      1010
## 60 7:30 PM      21      1010
```

```
barplot(t[,3], ylab = "température (°C)", xlab = "heures")
```



```
## [1] "Time"          "TimeCEST"
## [3] "TemperatureC"   "Dew_PointC"
## [5] "Humidity"       "Sea_Level_PressurehPa"
## [7] "VisibilityKm"   "Wind_Direction"
## [9] "Wind_SpeedKm_h" "Gust_SpeedKm_h"
## [11] "Precipitationmm" "Events"
## [13] "Conditions"     "WindDirDegrees"
## [15] "DateUTC"
```

4 Température du mois

```
library(xts)

start_date <- "2015-06-01"
```

```

end_date <- "2015-07-28"

# Strasbourg
station = "LFST"
tm <- getWeatherForDate(station, start_date=start_date, end_date=end_date, opt_all_columns = TRUE)

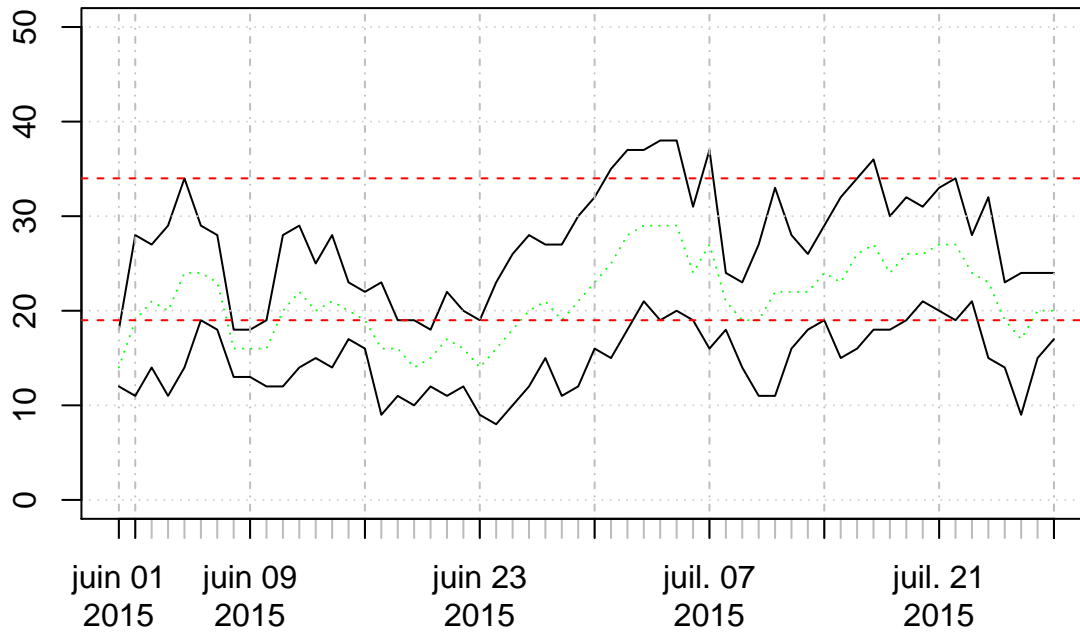
## [1] "CEST" "Max_TemperatureC"
## [3] "Mean_TemperatureC" "Min_TemperatureC"
## [5] "Dew_PointC" "MeanDew_PointC"
## [7] "Min_DewpointC" "Max_Humidity"
## [9] "Mean_Humidity" "Min_Humidity"
## [11] "Max_Sea_Level_PressurehPa" "Mean_Sea_Level_PressurehPa"
## [13] "Min_Sea_Level_PressurehPa" "Max_VisibilityKm"
## [15] "Mean_VisibilityKm" "Min_VisibilitykM"
## [17] "Max_Wind_SpeedKm_h" "Mean_Wind_SpeedKm_h"
## [19] "Max_Gust_SpeedKm_h" "Precipitationmm"
## [21] "CloudCover" "Events"
## [23] "WindDirDegrees"

## [1] "Date" "CEST"
## [3] "Max_TemperatureC" "Mean_TemperatureC"
## [5] "Min_TemperatureC" "Dew_PointC"
## [7] "MeanDew_PointC" "Min_DewpointC"
## [9] "Max_Humidity" "Mean_Humidity"
## [11] "Min_Humidity" "Max_Sea_Level_PressurehPa"
## [13] "Mean_Sea_Level_PressurehPa" "Min_Sea_Level_PressurehPa"
## [15] "Max_VisibilityKm" "Mean_VisibilityKm"
## [17] "Min_VisibilitykM" "Max_Wind_SpeedKm_h"
## [19] "Mean_Wind_SpeedKm_h" "Max_Gust_SpeedKm_h"
## [21] "Precipitationmm" "CloudCover"
## [23] "Events" "WindDirDegrees"

xt <- xts(tm[, c(3, 4, 5)], order.by = tm[,1])
plot(xt[,1], ylim = c(0,50), main = "Evolution des températures à Strasbourg")
lines(xt[,3], ylim = c(0,50))
lines(xt[,2], ylim = c(0,50), lty = 3, col = "green")
abline(h = 34, lty = 2, col = "red")
abline(h = 19, lty = 2, col = "red")

```

Evolution des températures à Strasbourg



```
# Mulhouse
station = "LFSB"
tm <- getWeatherForDate(station, start_date=start_date, end_date=end_date, opt_all_columns = TRUE)
```

```
## [1] "CEST" "Max_TemperatureC"
## [3] "Mean_TemperatureC" "Min_TemperatureC"
## [5] "Dew_PointC" "MeanDew_PointC"
## [7] "Min_DewpointC" "Max_Humidity"
## [9] "Mean_Humidity" "Min_Humidity"
## [11] "Max_Sea_Level_PressurehPa" "Mean_Sea_Level_PressurehPa"
## [13] "Min_Sea_Level_PressurehPa" "Max_VisibilityKm"
## [15] "Mean_VisibilityKm" "Min_VisibilityKm"
## [17] "Max_Wind_SpeedKm_h" "Mean_Wind_SpeedKm_h"
## [19] "Max_Gust_SpeedKm_h" "Precipitationmm"
## [21] "CloudCover" "Events"
## [23] "WindDirDegrees"

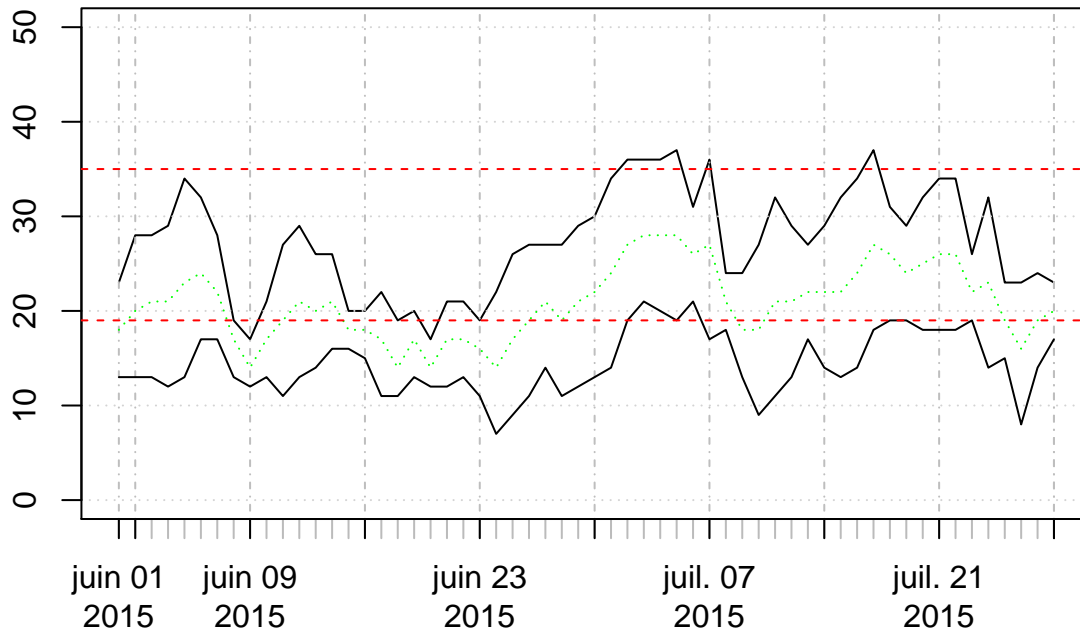
## [1] "Date" "CEST"
## [3] "Max_TemperatureC" "Mean_TemperatureC"
## [5] "Min_TemperatureC" "Dew_PointC"
## [7] "MeanDew_PointC" "Min_DewpointC"
## [9] "Max_Humidity" "Mean_Humidity"
## [11] "Min_Humidity" "Max_Sea_Level_PressurehPa"
## [13] "Mean_Sea_Level_PressurehPa" "Min_Sea_Level_PressurehPa"
## [15] "Max_VisibilityKm" "Mean_VisibilityKm"
## [17] "Min_VisibilityKm" "Max_Wind_SpeedKm_h"
## [19] "Mean_Wind_SpeedKm_h" "Max_Gust_SpeedKm_h"
## [21] "Precipitationmm" "CloudCover"
## [23] "Events" "WindDirDegrees"
```

```

xt <- xts(tm[, c(3, 4, 5)], order.by = tm[,1])
plot(xt[,1], ylim = c(0,50), main = "Evolution des températures à Mulhouse")
lines(xt[,3], ylim = c(0,50))
lines(xt[,2], ylim = c(0,50), lty = 3, col = "green")
abline(h = 35, lty = 2, col = "red")
abline(h = 19, lty = 2, col = "red")

```

Evolution des températures à Mulhouse



```

# Colmar
station = "LFSC"
tm <- getWeatherForDate(station, start_date=start_date, end_date=end_date, opt_all_columns = TRUE)

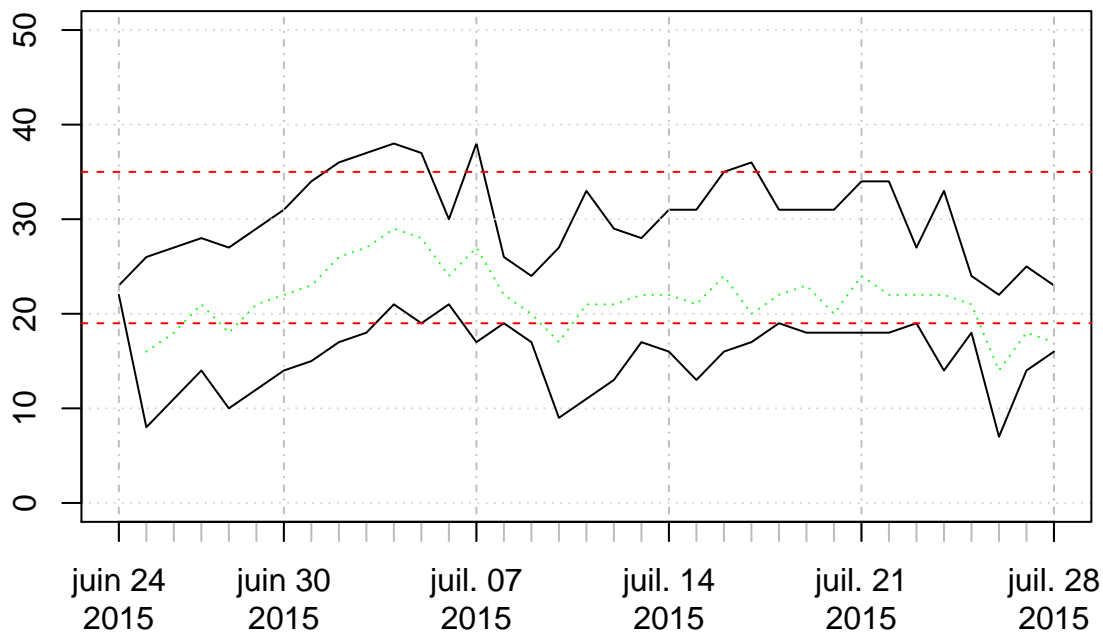
```

## [1] "CEST"	"Max_TemperatureC"
## [3] "Mean_TemperatureC"	"Min_TemperatureC"
## [5] "Dew_PointC"	"MeanDew_PointC"
## [7] "Min_DewpointC"	"Max_Humidity"
## [9] "Mean_Humidity"	"Min_Humidity"
## [11] "Max_Sea_Level_PressurehPa"	"Mean_Sea_Level_PressurehPa"
## [13] "Min_Sea_Level_PressurehPa"	"Max_VisibilityKm"
## [15] "Mean_VisibilityKm"	"Min_VisibilitykM"
## [17] "Max_Wind_SpeedKm_h"	"Mean_Wind_SpeedKm_h"
## [19] "Max_Gust_SpeedKm_h"	"Precipitationmm"
## [21] "CloudCover"	"Events"
## [23] "WindDirDegrees"	
## [1] "Date"	"CEST"
## [3] "Max_TemperatureC"	"Mean_TemperatureC"
## [5] "Min_TemperatureC"	"Dew_PointC"
## [7] "MeanDew_PointC"	"Min_DewpointC"
## [9] "Max_Humidity"	"Mean_Humidity"
## [11] "Min_Humidity"	"Max_Sea_Level_PressurehPa"


```
## [13] "Mean_Sea_Level_PressurehPa" "Min_Sea_Level_PressurehPa"
## [15] "Max_VisibilityKm"           "Mean_VisibilityKm"
## [17] "Min_VisibilitykM"           "Max_Wind_SpeedKm_h"
## [19] "Mean_Wind_SpeedKm_h"        "Max_Gust_SpeedKm_h"
## [21] "Precipitationmm"           "CloudCover"
## [23] "Events"                     "WindDirDegrees"
```

```
xt <- xts(tm[, c(3, 4, 5)], order.by = tm[,1])
plot(xt[,1], ylim = c(0,50), main = "Evolution des températures à Colmar")
lines(xt[,3], ylim = c(0,50))
lines(xt[,2], ylim = c(0,50), lty = 3, col = "green")
abline(h = 35, lty = 2, col = "red")
abline(h = 19, lty = 2, col = "red")
```

Evolution des températures à Colmar



5 Quand déclare-t-on l'état de canicule ?

Par définition, on parle de canicule lorsque les températures observées sont élevées jour et nuit pendant trois jours d'affilée. « Plus il fait chaud le jour, plus on fatigue. Plus il fait chaud la nuit, moins on récupère », explique Météo France.

L'organisme météorologique a déterminé des seuils d'alerte à partir de trente années de données quotidiennes de mortalité et de différents indicateurs météorologiques. Il les compare à des indicateurs biométéorologiques calculés à partir des températures minimales et maximales quotidiennes, en fonction de ses prévisions.

Les diverses régions de France étant plus ou moins habituées et donc adaptées à la chaleur, les seuils d'alertes ne sont pas les mêmes partout. Ainsi, Météo France estime que la canicule sera avérée dans les Deux-Sèvres lorsque des températures minimales de 35 °C le jour et 20 °C la nuit seront observées. La Haute-Loire, elle, sera considérée en canicule lorsque les températures dépasseront 32 °C le jour et 18 °C la nuit. A Paris, ces seuils sont de 31 °C le jour et 21 °C la nuit, ou encore à Marseille de 35 °C le jour et 24 °C la nuit.

En savoir plus sur http://www.lemonde.fr/les-decodeurs/article/2015/07/01/comment-le-plan-national-canicule-fonctionne-t-il_4665550_4355770.html#lswgUrOuW6g.99

Jour de canicule: 2015-07-04

Département	Jour	Nuit
Bas-Rhin	34	19
Haut-Rhin	35	19
Moselle	34	19
Vosges	34	18
Meurthe&M	34	18