Reproducible biometeorological research with R (an example)

Ioannis Charalampopoulos

22/06/2020

Introduction

The following is an absolutely hypothetical research scenario.

Scope

We want to investigate the thermal comfort over an area from Toulouse (France) to Andorra Principality.

Data, method and output

The data will be carried out from NASAPOWER data repository. The selection will be made by the definition of geographical coordinates (LAT/LON).

We will calculate the THI and Wind Chill indices for a ten year period (1995-2005) in a daily basis with rBiometeo package. The output will be a rendered as a .pdf file containing the text and the code in grey boxes,a violin-plot in .png format and and time-series facet plot in .jpeg format both of them in specific size (width/ height). Also, the data table will saved in a .csv file. Of, course the user can run the code directly by this .Rmd file to reproduce all the research workflow.

Data acquisition

We use the nasapower package to download air temperature (average, max, min) and Relative Humidity (RH) at 2m, and Wind Speed (WS) at 2 and 10 meters.

```
#Load package
library(nasapower)

# Retrieve data from NASAPOWER
my_data <- get_power(
  community = "AG",
  lonlat = c(0.83206,42.5431,2.08450,43.8485),
  pars = c("T2M","T2M_MAX","T2M_MIN","WS2M","WS10M", "RH2M"),
  dates = c("1995-01-01", "2005-12-31"),
  temporal_average = "DAILY"
)
```

Data handling

```
# Load package
library(dplyr)

# Convert the downloaded data to table
my_data_tibble <- as_tibble(my_data)

# Create a column with seasons
my_data_tb <- my_data_tibble %>%
  mutate(Season =
           ifelse(MM %in% c(1, 2, 12), "Winter",
                  ifelse(
                    MM %in% c(3, 4, 5), "Spring",
                    ifelse(MM %in% c(6, 7, 8), "Summer",
                           ifelse(MM %in% c(9, 10, 11), "Autumn", "Error"))
                  )))
```

Data analysis

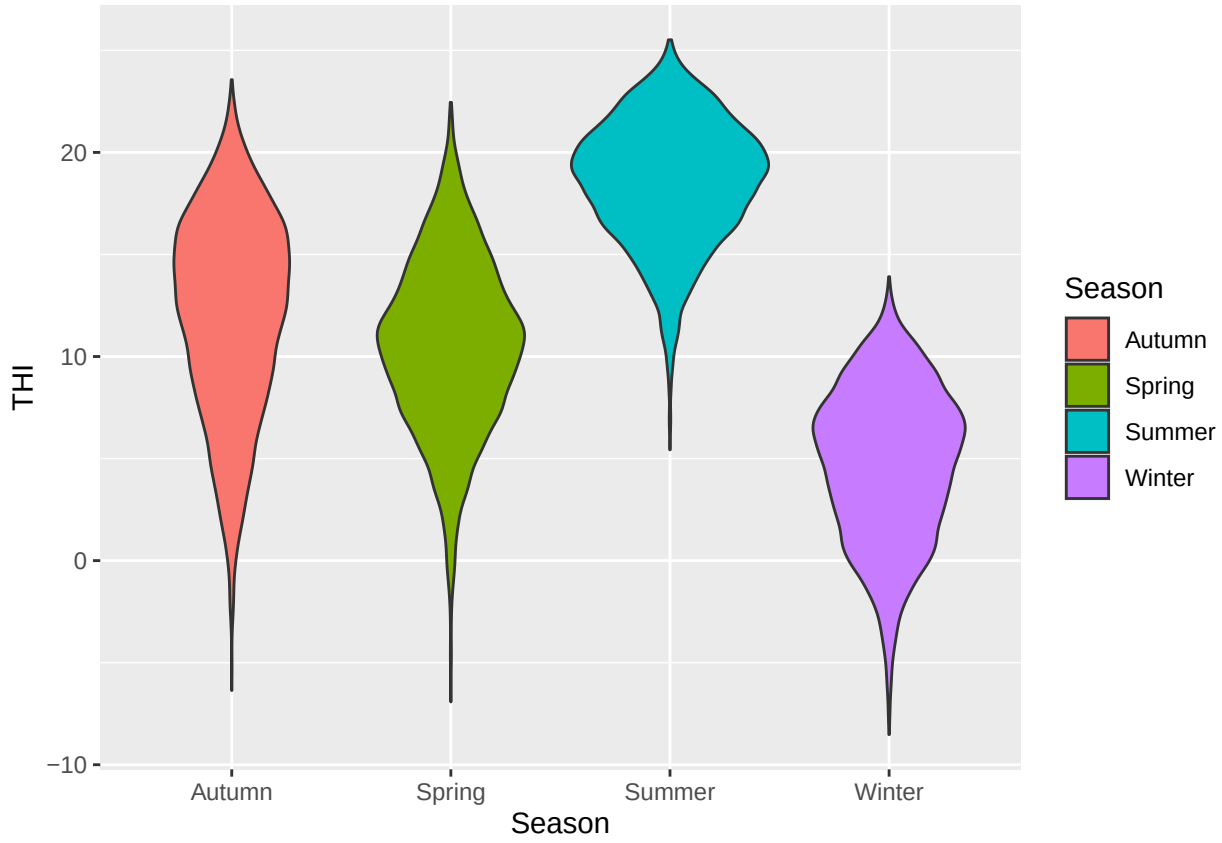
```
# Load packages
library(rBiometeo)
library(dplyr)

# Add a column with THI index
my_data_tb  <-  my_data_tb %>%
  mutate(THI = rBiometeo::THI(T2M, RH2M))

# Add a column with Wind chill index
my_data_tb  <-  my_data_tb %>%
  mutate(Windchill = rBiometeo::windchill(T2M, WS2M))

my_data_tb$YEAR <- as.integer(my_data_tb$YEAR)


# Inspect the first rows of the new dataset
head(my_data_tb)
```

```
## # A tibble: 6 x 16
##     LON   LAT  YEAR    MM    DD   DOY YYYYMMDD     T2M T2M_MAX T2M_MIN  WS2M
##   <dbl> <dbl> <int> <int> <int> <int> <date>     <dbl>   <dbl>   <dbl> <dbl>
## 1  0.75  42.8  1995     1     1     1 1995-01-01 -1.86    2.56   -5.38  3.18
## 2  1.25  42.8  1995     1     1     1 1995-01-01 -1.9     2.44   -5.37  2.28
## 3  1.75  42.8  1995     1     1     1 1995-01-01 -1.84    2.56   -5.53  1.97
## 4  2.25  42.8  1995     1     1     1 1995-01-01  0.26    3.85   -3.31  2.4
## 5  0.75  43.2  1995     1     1     1 1995-01-01  1.57    4.21   -1.03  2.86
## 6  1.25  43.2  1995     1     1     1 1995-01-01  1.83    4.42   -0.78  2.2
## # ... with 5 more variables: WS10M <dbl>, RH2M <dbl>, Season <chr>, THI <dbl>,
## #   Windchill <dbl>
```

Data dissemination (results visualisation)

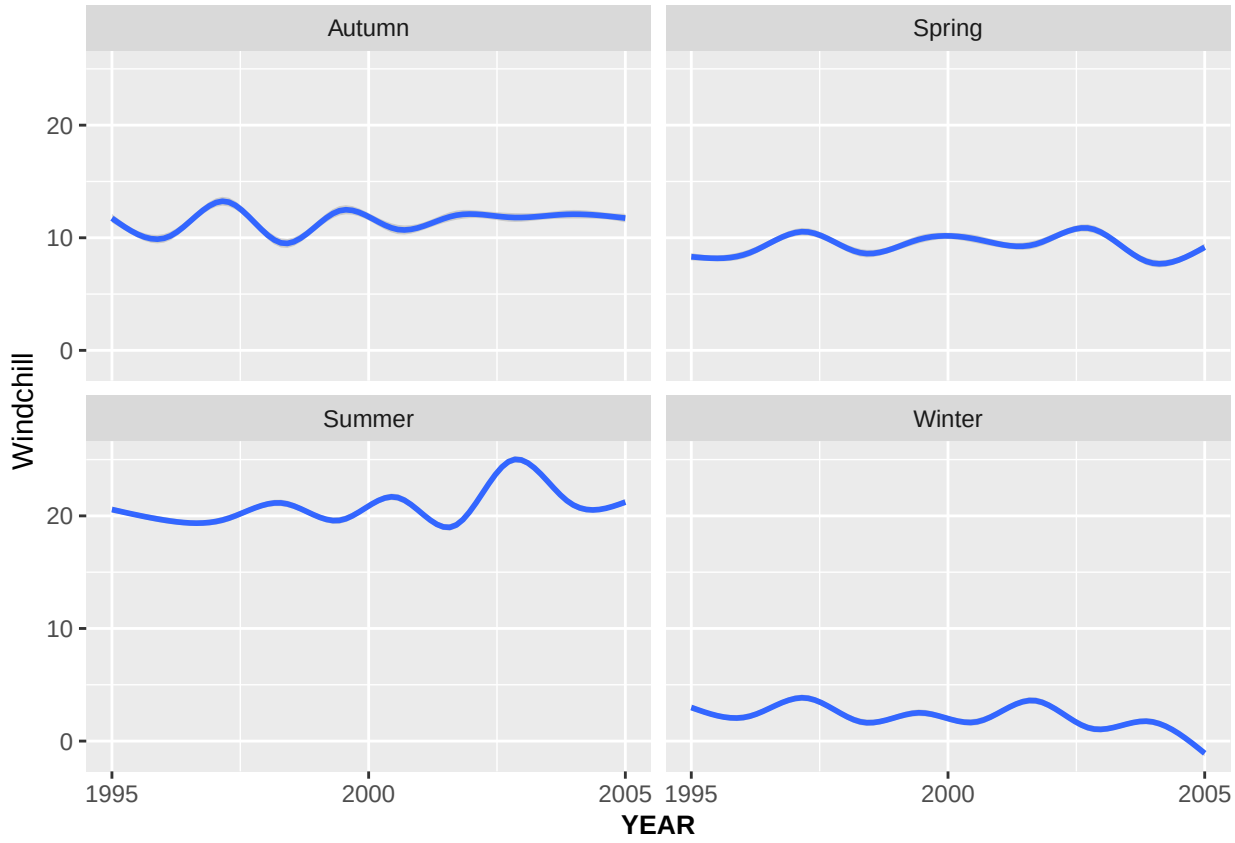
The results of the indices' calculations will be saved as an .csv file and will be used for two graphs (violin-plot and time series plot with facets)

```
# Create a violin plot for the THI index for all sampling points per season
# and save it in png format with specific size

## Load the library
library(ggplot2)


## Create the violin plot
violin_plot <-
  ggplot(my_data_tb, aes(x = Season, y = THI, fill = Season)) +
  geom_violin()

## Illustrate the violin plot
plot(violin_plot)
```




```
## Save the last plot in a .png file with the below size
ggsave(
  "violin_plot.png",
  width = 40,
  height = 20,
  units = "cm"
)
```

```r
## Create smoothed line graphs of the Windchill index for all the sampling
## points per season
smooth_plot <- ggplot(my_data_tb, aes(x = YEAR, y = Windchill)) +
  geom_smooth() +
  facet_wrap(. ~ Season) +
  scale_x_continuous(breaks = c(1995, 2000, 2005)) +
  theme(axis.title.x = element_text(color = "black", size = 10, face = "bold"))

plot(smooth_plot)
```




```r
## Save the last plot in a .jpeg file with the below size and resolution
ggsave(
  "time_plot.jpeg",
  width = 40,
  height = 20,
  units = "cm",
  dpi = 600
)


## Save the data table in a .csv file called indices.csv
write.csv(my_data_tb, file = "indices.csv")
```