Tasks for chapter 7: Going beyond linearity: single stations

Daily average 12 UTC temperature of a SYNOP station in Austria

This time use only the SYNOP observation of temperature `t2mObs` for the "location" assigned to your dyad, contained in `location_obs_ECMWF_2009-2022.rds`. In the following text, replace "location" with the name of the one assigned to you, e.g. "ibk". Convert only the column `t2mObs` for the period 2009-2021 to a data frame and assign to `locationT` using `as.data.frame()`.

Add the date information from the zoo-object `location` (contained in `index(location)`) to `locationT$date`.

Compute the sequential numbers of each day in the year (sometimes called "Julian days") for each date with `locationT$yday <- as.POSIXlt(locationT$date)$yday`. Note that the numbering starts with zero!!!

Check whether there are any days without temperatures with `which(is.na(locationT$T)` and if so eliminate these days (`na.omit`).

Make all years 365 days long by dropping the last day of a leap year using `subset()`.

Plot the temperature time series as points without connecting lines.

Compute daily average temperatures `Tmean` and daily median temperatures `Tmedian` and plot both as points for days 1 ... 365. Use `aggregate()`. For example, `Tmean <- aggregate(locationT$T, by = list(locationT$yday), FUN = mean)`. `FUN` can be any function in R or also a user-supplied function. Plot daily temperatures, daily mean and daily median temperatures as function of day of year. Explain the smoothness (or its lack) of the annual course of these 3 temperatures.

Compute a running mean of the daily averages and running median of daily median temperatures, respectively, using a 15-day filter and plot it as lines onto the previous figure.

Compute a local regression smoothing to get daily average temperature using `loess()` with default options and add the line to the previous ones.

Compute smooth daily average temperatures using a GAM with a cyclic spline (`bs = "cc"`). Use the `mgcv` package and the command `fitG <- gam(T ~ s(yday, bs = "cc"), data = locationT)`. What are the effective degrees of freedom? How large is the intercept?

Plot the GAM curve with intercept added (since the smooth effects are centered around zero), which can be achieved with `plot.gam(fitG, shift  = fitG$coefficients[1])`, daily average temperatures as points, 15-day running mean curve and LOESS curve and comment on differences and your preferred choice.

Daily average precipitation amount for a Tyrolean weather station

Perform the same tasks as for daily temperature in Innsbruck but for daily precipitation sums for the station contained in the `*.rda` file assigned to your dyad. Remember to convert from zoo-object to a data frame. Only difference to temperature:
power-transform the precipitation amounts with an exponent of 1/1.6. And don't forget to re-transform them for final plot with GAM, moving average, local-regression fit and data points as function of day of year.

Predicting temperature with a GAM

Use the `location_obs_ECMWF_2009-2022.rds` data for the "location" assigned to your dyad. Leave the period from 2020-01-01 till the end of the series as test period and train a GAM using the mgcv-package on the remaining data, assigning it to `gmod`. Remember to drop missing values, convert to a data frame, and compute yday etc - as in the first task. Use a smooth term each for `t2m` and `lcc` ECMWF variables with a basis `bs = "cr"` and a cyclic smooth term for yday with basis `bs = "cc"`. Plot the effects of these terms using `plot(gmod)`. Note that all effects are centered.

Use `predict()` with newdata from the test period and plot the time series of differences between GAM-forecasted and observed temperatures.

Then fit a GAM with an additional interaction term between `t2m` and `yday`, using the tensor product `ti(t2m, yday, bs = c("cr", "cc"))` and again plot the model and the differences of forecasted minus observed for the test period. Interpret any differences to the previous method.