Chapter 8: Random forests – predicting ozone concentration

We move on from a single tree to an ensemble of trees - a forest. The dataset remains the same and also the package `partykit` to fit the forest to the data.

1. Data preparation

- The ozone measurements are in the package `datasets` and can be loaded with `data("airquality")`.
- Convert `Month` and `Day` to day of year with `as.POSIXlt(paste(1973, airquality$Month, airquality$Day), format = "%Y %m %d")$yday`
- Convert the units to SI: Langley to Joule per square meter, miles per hour to meters per second, and Fahrenheit to Celsius.
- Subset to non-missing ozone data (other variables can be missing) and drop `Month` and `Day`, using `subset()` and assign to `airq`. (Dropping works with the option `select = -c(Month, Day)`).

2. Fit forest

- Set a seed of 2908 to make the results reproducible
- Fit a conditional random forest consisting of `ntrees = 100` trees with `partykit::cforest(formula, data = ...)` using all variables and assign to `airQcforest`. For the usual formula syntax, remember the abbreviation ".", which uses all variables in the data set except for the response variable to the left of the "~" symbol. (In case you're wondering why the package is explicitly specified: it makes absolutely sure that you don't get `cforest` from the `party` package.)

3. Examine and plot the resulting forest model

"Random forests are a widely used ensemble learning method for classification or regression tasks. However, they are typically used as a black box prediction method that offers only little insight into their inner workings. ... the `stablelearner` package can be used to gain insight into this black box by visualizing and summarizing the variable and cutpoint selection of the trees within a random forest." (Quote from the vignette https://cran.r-project.org/web/packages/stablelearner/vignettes/forests.html). Reading the vignette will help you with the interpretation asked of you in the following parts.

- Load the `stabletree` package.
- Convert the conditional random forest model in `airQcforest` to a stabletree object with `as.stabletree(airQcforest)` and assign to `airQcforest_st`.
- Do a summary of `airQcforest_st` and interpret.
- Use a `barplot()` to show how frequently the predictors were selected in each of the `ntrees` and interpret.
- Use `image()` to show which variables were selected when a particular variable was NOT selected and interpret.
- Use `plot()` to inspect the cutpoints and resulting partitions for each variable over all `ntrees`. Which predictor variables are well suited for random forests and for which ones might a simple (generalized) linear regression have been fine, too?