Plotting with openair

The openair package is specifically designed to plot air pollution data. This tutorial will give a brief introduction to many of the plotting functions in openair. Visit the project website for more information, including a comprehensive manual.

This tutorial will cover the following openair functions.

* [summaryPlot()](#summary)
* [windRose()](#windrose)
* [pollutionRose()](#pollutionrose)
* [timePlot()](#timeplot)
* [calendarPlot()](#calendarplot)

## summaryPlot()

The first plotting function we'll look at is summaryPlot(). Functions in the openair package expect data frames that are in a certain format. The column with date and time information must be labeled "date" (lowercase) and be a POSIXct class.

The first data frame we'll use is the chicago\_air data frame from the region5air package, which already has a column labeled "date".

library(region5air)  
data(chicago\_air)  
head(chicago\_air)

## date ozone temp solar month weekday  
## 1 2013-01-01 0.032 17 0.65 1 3  
## 2 2013-01-02 0.020 15 0.61 1 4  
## 3 2013-01-03 0.021 28 0.17 1 5  
## 4 2013-01-04 0.028 18 0.62 1 6  
## 5 2013-01-05 0.025 26 0.48 1 7  
## 6 2013-01-06 0.026 36 0.47 1 1

However, the class of that "date" column is character.

class(chicago\_air$date)

## [1] "character"

We need to change it using the as.POSIXct() function. This function will assume that the format of the date is YYYY-MM-DD. So we just need to supply a correct time zone.

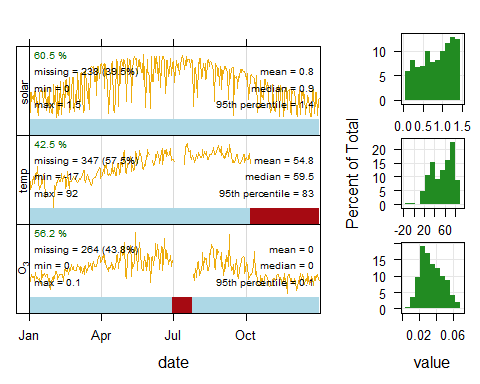
chicago\_air$date <- as.POSIXct(chicago\_air$date, tz = "America/Chicago")

If your date (and time) isn't in the default format, see the "Working with Dates in R" section of this tutorial. If you need to provide a different time zone, run the OlsonNames() function for a list of options.

Now we just feed the first four columns of the data frame to the summaryPlot() function. We use the select() function from dplyr to select our columns, and we use the short hand date:solar to select the "date" column, the "solar" column, and all the columns in between.

library(openair)  
library(dplyr)  
  
summaryPlot(select(chicago\_air, date:solar))

## date1 date2 ozone temp solar   
## "POSIXct" "POSIXt" "numeric" "numeric" "numeric"



The first column of the graph contains time series plots of all of the parameters in the data frame. The red bars at the bottom of each panel show where there are large segments of missing data. The panels on the right are the histograms of the distributions for each parameter.

## windRose()

The windRose() function expects a data frame with columns for wind speed and wind direction labeled "ws" and "wd", respectively. Here we load the chicago\_wind dataset from the region5air package and take a look at the columns.

data(chicago\_wind)  
head(chicago\_wind)

## Source: local data frame [6 x 4]  
## Groups: datetime [6]  
##   
## datetime wind\_speed wind\_dir ozone  
## (chr) (dbl) (dbl) (dbl)  
## 1 20130101T0000-0600 2.0 334 NA  
## 2 20130101T0100-0600 2.0 321 NA  
## 3 20130101T0200-0600 2.0 323 NA  
## 4 20130101T0300-0600 1.6 324 NA  
## 5 20130101T0400-0600 1.6 319 NA  
## 6 20130101T0500-0600 1.4 319 NA

We need to create a "date" column with a POSIXct class, so we'll use the as.POSIXct() function again. This time we will need to provide information in the format parameter.

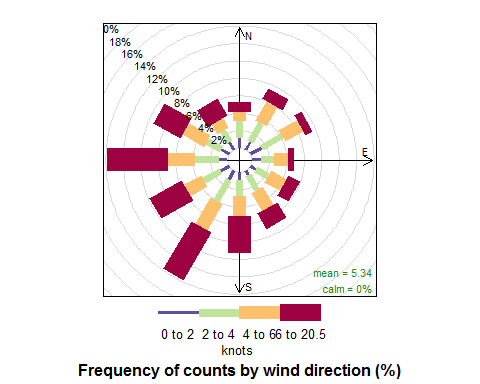
chicago\_wind$datetime <- as.POSIXct(chicago\_wind$datetime, format ="%Y%m%dT%H%M",  
 tz = "America/Chicago")

Now we'll rename the columns using the rename() function from dplyr.

chicago\_wind <- rename(chicago\_wind, date = datetime, ws = wind\_speed, wd = wind\_dir)

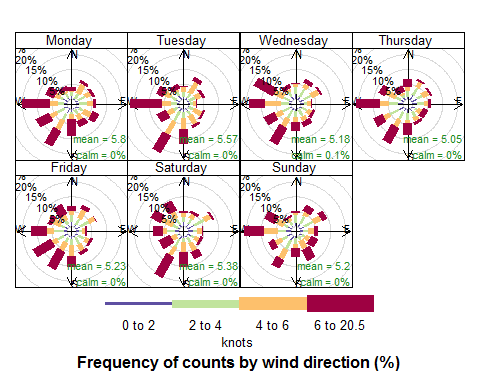
Then we can feed the data frame to the windRose() function.

windRose(chicago\_wind, key.footer = "knots") # default is m/s



We can split the data frame by time periods by using the type argument.

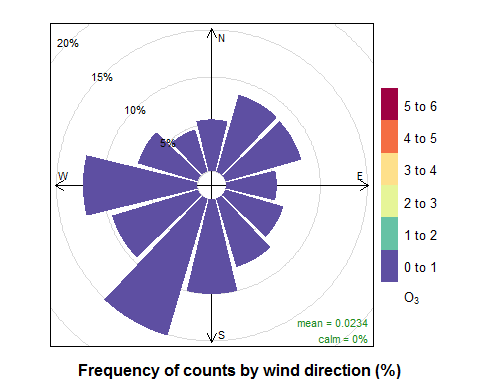
windRose(chicago\_wind, type = "weekday", key.footer = "knots")



## pollutionRose()

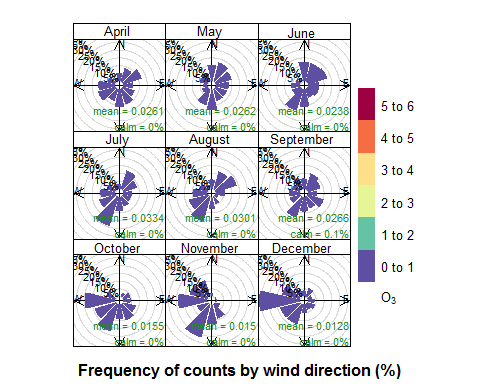
We can make a similar plot that will display pollutant concentrations in relation to wind direction.

pollutionRose(chicago\_wind, pollutant = "ozone")



We can also look at the values by time periods.

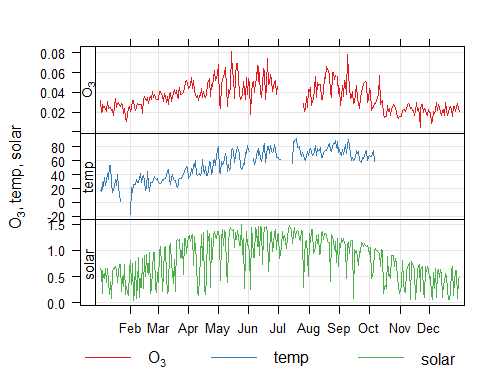
pollutionRose(chicago\_wind, pollutant = "ozone", type = "month")



## timePlot()

Time series plots can be easily produced using timePlot()

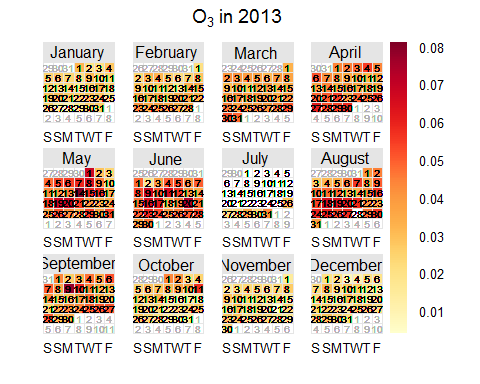
timePlot(chicago\_air, pollutant = c("ozone", "temp", "solar"))



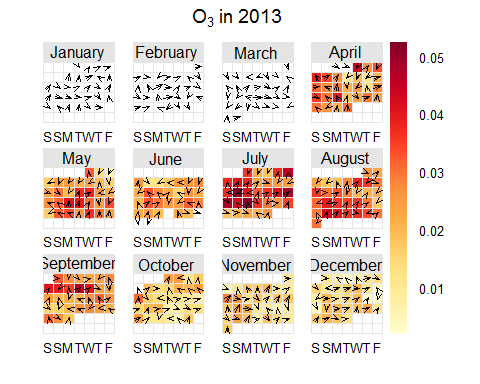
## calendarPlot()

calendarPlot() displays daily values in a calendar format.

calendarPlot(chicago\_air, pollutant = "ozone")



calendarPlot(chicago\_wind, pollutant = "ozone", annotate = "ws")



## Exercises

Go to the exercises page for this tutorial and test your comprehension of this material.