# Openair R Library Notes

Authors: Monika Roznere and Andy Bean

User friendly documentation: <https://bookdown.org/david_carslaw/openair/>

Github repo: <https://github.com/davidcarslaw/openair>

ToC:

[Openair R Library Notes](#_nanzayn0kj6t)

[Simple example of loading and plotting data from openair git repo.](#_8kit3z18cxt0)

[Examples of openair functionalities.](#_4y3cpopchi2h)

[Utility Functions](#_bk2naw5i1q2y)

To install openair on RStudio, type:

install.packages(“openair”)

Then execute, and openair along with other necessary packages will begin downloading immediately

## 

## Simple example of loading and plotting data from openair git repo.

This is provided in the ReadMe of the openair git repository above.

“To import hourly data from 100s of sites and to import several sites at one time and several years of data.” (Importing data from the UK Automatic Urban and Rural Network.)

library(openair)

kc1 <- importAURN(site = "kc1", year = 2011:2012)

head(kc1)

## # A tibble: 6 x 18

## site code date co nox no2 no o3 so2 pm10

## <chr> <fct> <dttm> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 Lond… KC1 2011-01-01 00:00:00 0.2 44 38 4 14 5 40

## 2 Lond… KC1 2011-01-01 01:00:00 0.2 38 29 6 28 3 36

## 3 Lond… KC1 2011-01-01 02:00:00 0.2 32 31 1 18 3 31

## 4 Lond… KC1 2011-01-01 03:00:00 0.2 31 29 1 14 3 31

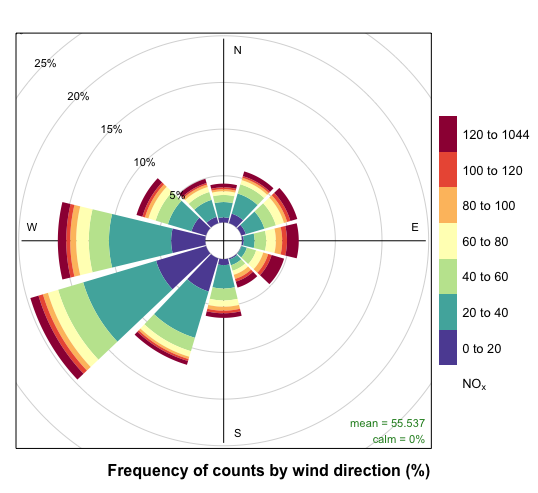
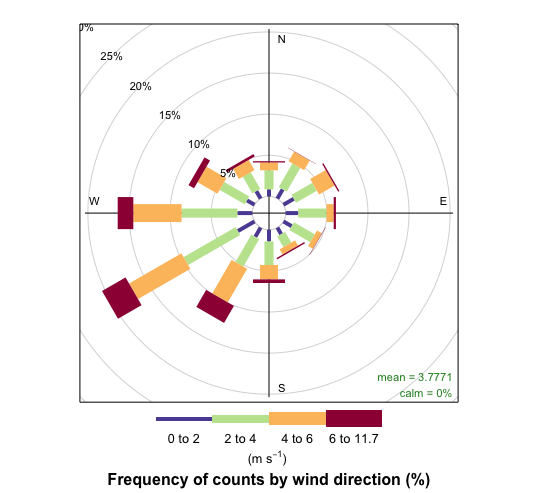
## 5 Lond… KC1 2011-01-01 04:00:00 0.2 31 29 1 16 3 29

## 6 Lond… KC1 2011-01-01 05:00:00 0.1 29 27 1 24 3 25

## # … with 8 more variables: pm2.5 <dbl>, v10 <dbl>, v2.5 <dbl>, nv10 <dbl>,

## # nv2.5 <dbl>, ws <dbl>, wd <dbl>, air\_temp <dbl>

windRose(kc1)



Using the selectByDate function it is easy to select quite complex time-based periods. For example, to select weekday (Monday to Friday) data from June to September for 2012 and for the hours 7am to 7pm inclusive:

sub <- selectByDate(kc1, day = "weekday", year = 2012, month = 6:9, hour = 7:19)

head(sub)

## # A tibble: 6 x 18

## date site code co nox no2 no o3 so2 pm10

## <dttm> <chr> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>

## 1 2012-06-01 07:00:00 Lond… KC1 0.23 36 23 9 24 3 6

## 2 2012-06-01 08:00:00 Lond… KC1 0.23 33 21 7 34 3 9

## 3 2012-06-01 09:00:00 Lond… KC1 0.23 23 19 2 52 3 6

## 4 2012-06-01 10:00:00 Lond… KC1 0.23 17 13 2 62 3 7

## 5 2012-06-01 11:00:00 Lond… KC1 0.23 17 13 2 70 3 9

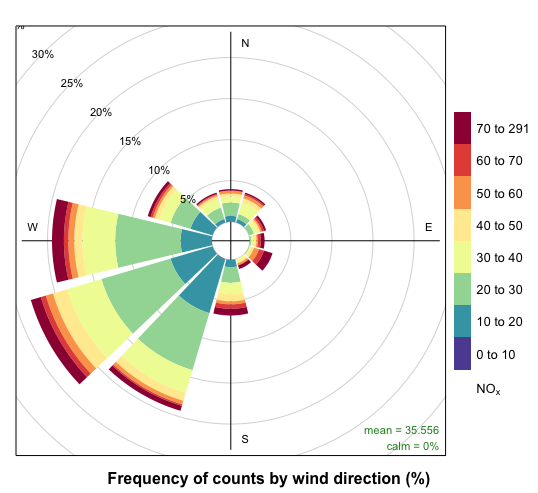
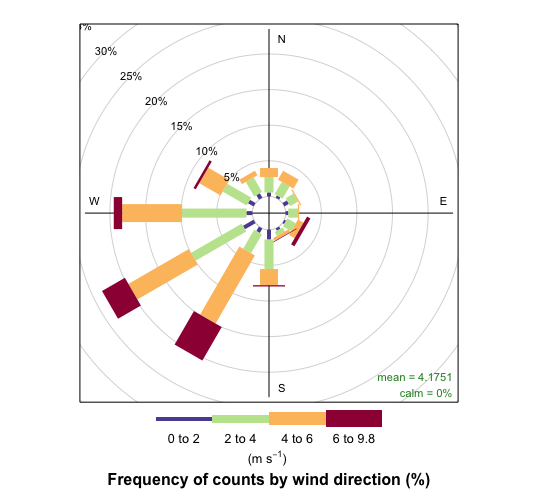
## 6 2012-06-01 12:00:00 Lond… KC1 0.23 21 19 1 78 3 8

## # … with 8 more variables: pm2.5 <dbl>, v10 <dbl>, v2.5 <dbl>, nv10 <dbl>,

## # nv2.5 <dbl>, ws <dbl>, wd <dbl>, air\_temp <dbl>

windRose(sub)

pollutionRose(sub)

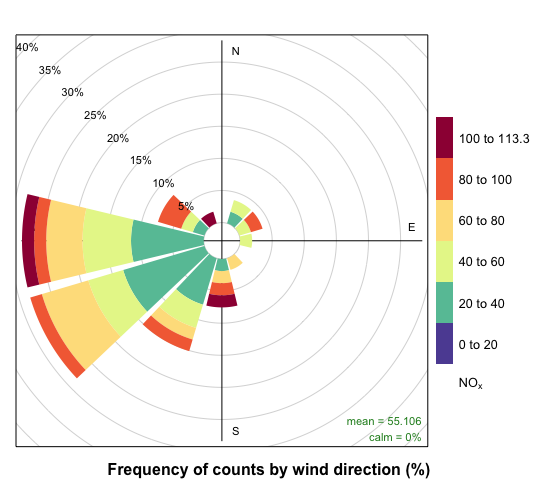
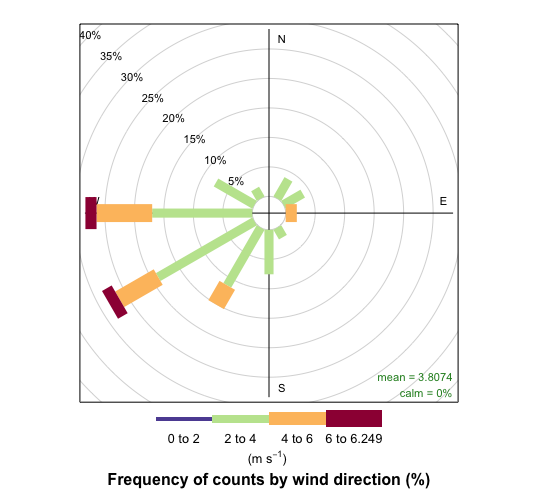


Similarly it is easy to time-average data in many flexible ways. For example, 2-week means can be calculated as:

sub2 <- timeAverage(kc1, avg.time = "2 week")

windRose(sub2)

pollutionRose(sub2)



## Examples of openair functionalities.

If you want to know required/optional arguments for a function, try: help(function\_name). For example, help(windRose).

| **Wind Rose**  <https://bookdown.org/david_carslaw/openair/sec-windRose.html>  “The windRose function can plot wind roses in a variety of ways: summarising all available wind speed and wind direction data, plotting individual wind roses by year, and also by month.”  windRose(mydata)  …and many more optional arguments. |  |
| --- | --- |
| **Pollution Rose**  <https://bookdown.org/david_carslaw/openair/sec-windRose.html>  “pollutionRose is a variant of windRose that is useful for considering pollutant concentrations by wind direction, or more specifically the percentage time the concentration is in a particular range.”  pollutionRose(mydata, pollutant = "nox")  …and many more optional arguments.  “The pollutionRose function is also useful for comparing two meteorological data sets.” |  |
| **Polar Frequencies**  <https://bookdown.org/david_carslaw/openair/sec-polarFreq.html>  “It is similar to the traditional wind rose, but includes a number of enhancements to also show how concentrations of pollutants and other variables vary.”  polarFreq(mydata)  …and many more optional arguments. |  |
| **Percentile Roses**  <https://bookdown.org/david_carslaw/openair/sec-percentileRose.html>  “percentileRose calculates percentile levels of a pollutant and plots them by wind direction.”  percentileRose(mydata)  …and many more optional arguments.  It can also plot conditional probability functions (CPF). |  |
| **Polar Plots**  <https://bookdown.org/david_carslaw/openair/sec-polarPlot.html>  “The polarPlot function plots a bivariate polar plot of concentrations. Concentrations are shown to vary by wind speed and wind direction. In many respects they are similar to the plots shown in [Polar Frequencies] but include some additional enhancements.”  polarPlot(mydata) … and more  Other options:   * Nonparametric Wind Regression * Conditional Probability Function plot * Pairwise statistics * Comparing two time periods * Clustering   + Clustering concentrations   + Clustering differences in concentrations * Polar plots on an interactive map\*\*\* |  |
| **Polar Annulus**  <https://bookdown.org/david_carslaw/openair/sec-polarAnnulus.html>  “The polarAnnulus function provides a way in which to consider the temporal aspects of a pollutant concentration by wind direction. This is another means of visualising diurnal, day of week, seasonal and trend variations. Plotting as an annulus, rather than a circle avoids to some extent the difficulty in interpreting values close to the origin.”  polarAnnulus(mydata) … and more |  |
| **Time Series Plots**  <https://bookdown.org/david_carslaw/openair/sec-timePlot.html>  “The timePlot function is designed to quickly plot time series of data, perhaps for several pollutants or variables.”  timePlot(mydata) … and more |  |
| **Temporal Variations**  <https://bookdown.org/david_carslaw/openair/sec-timeVariation.html>  “In air pollution, the variation of a pollutant by time of day and day of week can reveal useful information concerning the likely sources. For example, road vehicle emissions tend to follow very regular patterns both on a daily and weekly basis. By contrast some industrial emissions or pollutants from natural sources (e.g. sea salt aerosol) may well have very different patterns.”  timeVariation(mydata) … and more |  |
| **Time Proportion Plots**  <https://bookdown.org/david_carslaw/openair/sec-timeProp.html>  “The timeProp (‘time proportion’) function shows time series plots as stacked bar charts.”  ??? |  |
| **Trend Heat Maps**  <https://bookdown.org/david_carslaw/openair/sec-trendLevel.html>  “The trendLevel function provides a way of rapidly showing a large amount of data in a condensed way. It is particularly useful for plotting the level of a value against two categorical variables.”  trendLevel(mydata) … and more |  |
| **Calendar Plots**  <https://bookdown.org/david_carslaw/openair/sec-calendarPlot.html>  “The calendarPlot function provides an effective way to visualize data in this way by showing daily concentrations laid out in a calendar format.”  calendarPlot(mydata) … and more |  |
| **Theil-Sen Trends**  <https://bookdown.org/david_carslaw/openair/sec-TheilSen.html>  TheilSen(mydata) … and more |  |
| **Smooth Trends**  <https://bookdown.org/david_carslaw/openair/sec-smoothTrend.html>  smoothTrend(mydata) … and more  Options:   * Seasonal averages |  |
| **Scatter Plots**  <https://bookdown.org/david_carslaw/openair/sec-scatterPlot.html>  scatterPlot(mydata) … and more |  |
| **Trajectory Analysis**  <https://bookdown.org/david_carslaw/openair/sec-trajPlot.html>  traj <- importTraj(site = "london", year = 2010)  head(traj)  trajPlot(selectByDate(traj, start = "15/4/2010",  end ="21/4/2010"), map.cols =  openColours("hue", 10), col = "grey30")  Options:   * Trajectory gridded frequencies * Trajectory source contribution functions   + Identifying the contribution of high concentration back trajectories   + Allocating trajectories to different wind sectors * Potential Source Contribution Function (PSCF) * Concentration Weighted Trajectory (CWT) * Trajectory clustering |  |
| **Conditional Quantiles**  <https://bookdown.org/david_carslaw/openair/sec-conditionalQuantile.html>  “The conditional quantile plot differs from the quantile-quantile plot (Q-Q plot) that is often used to compare observations and predictions. A Q-Q~plot separately considers the distributions of observations and predictions, whereas the conditional quantile uses the corresponding observations for a particular interval in the predictions.” |  |
| **Model Evaluation**  <https://bookdown.org/david_carslaw/openair/sec-modStats.html>  “The modStats function provides key model evaluation statistics for comparing models against measurements and models against other models.” |  |
| **Taylor Diagram**  <https://bookdown.org/david_carslaw/openair/sec-TaylorDiagram.html>  “The diagram provides a way of showing how three complementary model performance statistics vary simultaneously. These statistics are the correlation coefficient R, the standard deviation (sigma) and the (centred) root-mean-square error.” |  |

## Utility Functions

<https://bookdown.org/david_carslaw/openair/sec-utility.html>

Functions or how to’s available in openair library:

* Selecting data by date
* Making intervals — cutData
* Selecting run lengths of values above a threshold — pollution episodes
* Calculating rolling means
* Aggregating data by different time intervals
* Calculating percentiles
* Correlation matrices

**Saqgetr**

To access the data for all of Europe, we need an additional library:

*install.packages(“saqgetr”)*

Call this library and the library dplyr

Follow the documentation for the creator of this function directly to download data for thousands of sites in Europe and at least several hundred in Germany: <https://github.com/skgrange/saqgetr>