

# Plotting with openair: Exercises

These exercises accompany the Plotting with `openair` tutorial: <http://rpubs.com/NateByers/Openair>. These exercises use data frames from the `region5air` package. Run the following code to clean out your global environment and load the data you need:

```
rm(list = ls())
library(region5air)
library(openair)
library(dplyr)
library(tidyr)
data(chicago_air)
data(chicago_wind)
data(airdata)
```

## Exercises

1. Create a properly formatted “date” column in the `chicago_wind` dataset. Use the `as.POSIXct()` function to make it a `POSIXct` class, and use the `rename()` function to rename the “datetime” column to “date”. Once you have created a properly formatted “date” column, run this filter on the data frame to remove one row with an NA in the “date” column:

```
chicago_wind <- filter(chicago_wind, !is.na(date))
```

**Note:** One hour was not formatted as a `POSIXct` class because of the switch to daylight savings time.

### Solution 1

2. Use the `summaryPlot()` function to visualize the `chicago_wind` dataset.

### Solution 2

3. Use the `windRose()` function on the `chicago_wind` dataset and split the data into different panels by season. Remember to rename the “wind\_speed” and “wind\_dir” columns as “ws” and “wd” respectively.

### Solution 3

4. Use the `pollutionRose()` function on the ozone data in the `chicago_wind` data frame and change the `statistic` parameter to “prop.mean”.

### Solution 4

---

## Advanced Exercise

5. Use the `filter()` function to subset the `airdata` data frame down to the site “840180890022”. Use the `group_by()` function to group by the “datetime” and “parameter” columns. Use `summarize()` to replace the “value” column with the mean for multiple values per hour/parameter (i.e., for sites with more than one poc). Use `tidyr` to reshape the data to a wide format. Make time series plots of the parameters using the `timePlot()` function in `openair`. Be sure to rename the columns and format the date column properly. **Hints:** Remember to remove rows that have an NA in the date-time column. Also, the `rename()` function will not work on this data frame, so use `names() <-` instead.

### Solution 5

---

## Solutions

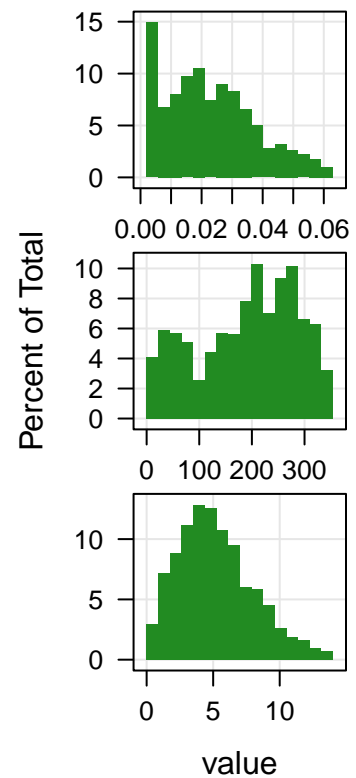
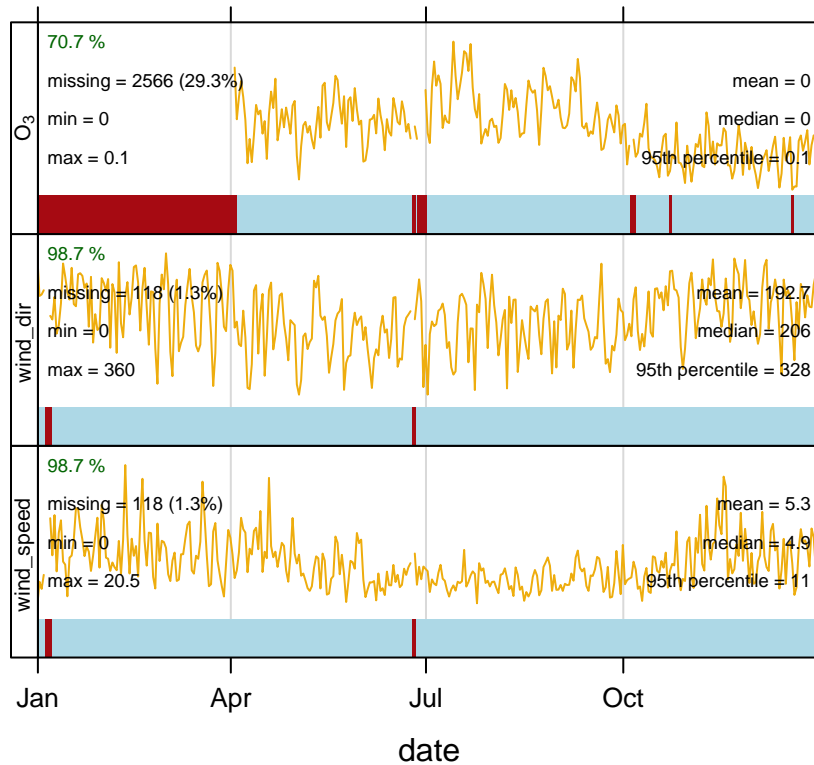
```
chicago_wind$datetime <- as.POSIXct(chicago_wind$datetime, format = "%Y%m%dT%H%M",
                                     tz = "America/Chicago")
chicago_wind <- rename(chicago_wind, date = datetime)
chicago_wind <- filter(chicago_wind, !is.na(date))
```

### Solution 1 [Back to exercises](#)

```
summaryPlot(chicago_wind)
```

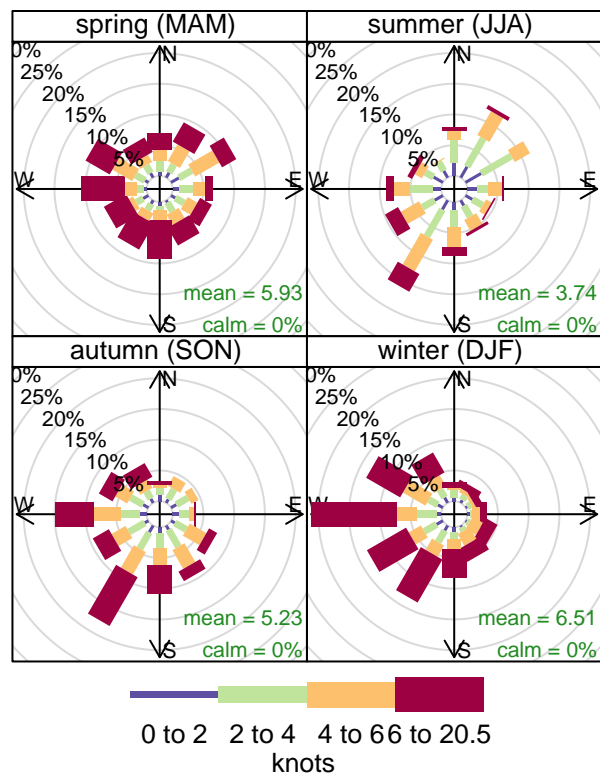
### Solution 2

```
##      date1      date2 wind_speed  wind_dir      ozone
## "POSIXct" "POSIXt"  "numeric"  "numeric"  "numeric"
```



[Back to exercises](#)

```
chicago_wind <- rename(chicago_wind, ws = wind_speed, wd = wind_dir)
windRose(chicago_wind, type = "season", key.footer = "knots")
```

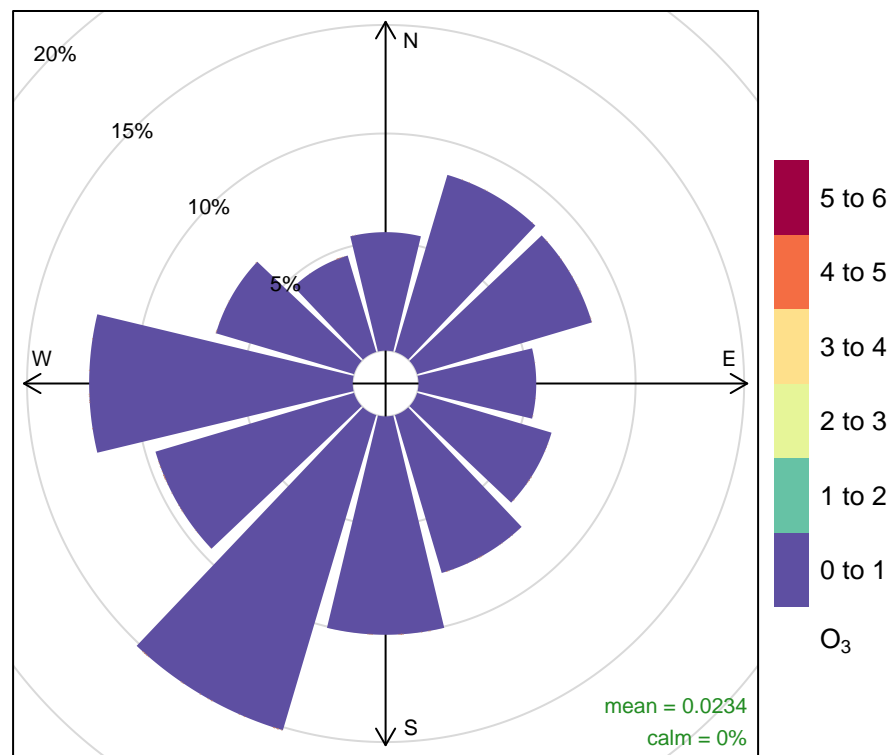


Solution 3

[Back to exercises](#)

**Solution 4** First let's plot with the default statistic of "prop.count".

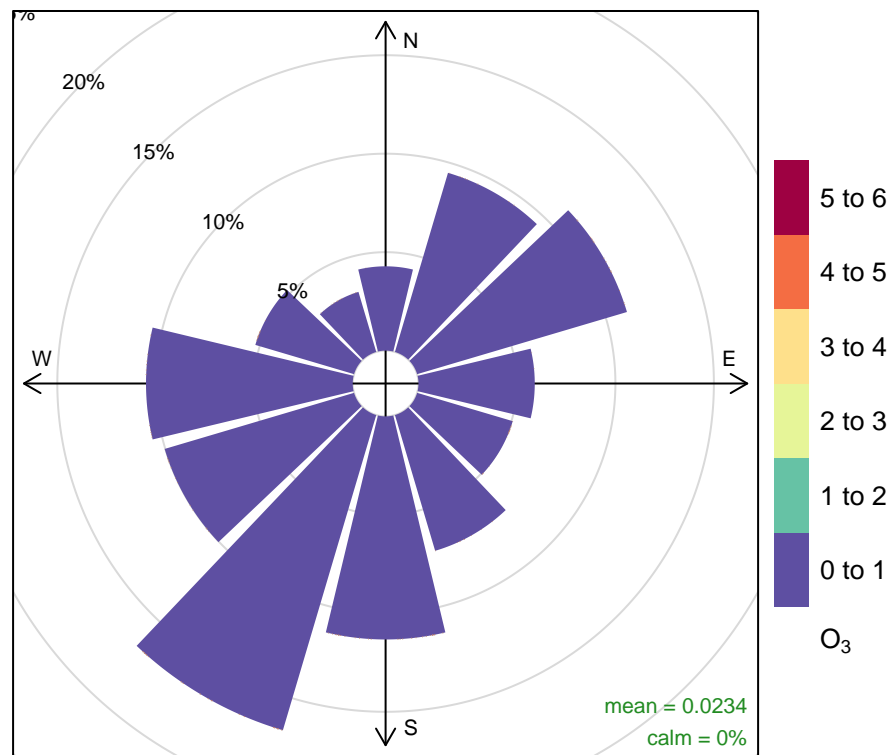
```
pollutionRose(chicago_wind, pollutant = "ozone", statistic = "prop.count")
```



**Frequency of counts by wind direction (%)**

Now we'll change it to "prop.mean".

```
pollutionRose(chicago_wind, pollutant = "ozone", statistic = "prop.mean")
```



**Proportion contribution to the mean (%)**

[Back to exercises](#)

```
# filter down to the right monitor and get the mean for multiple pocs
site22 <- filter(airdata, site == "840180890022")
site22 <- group_by(site22, datetime, parameter)
site22 <- summarize(site22, value = mean(value))

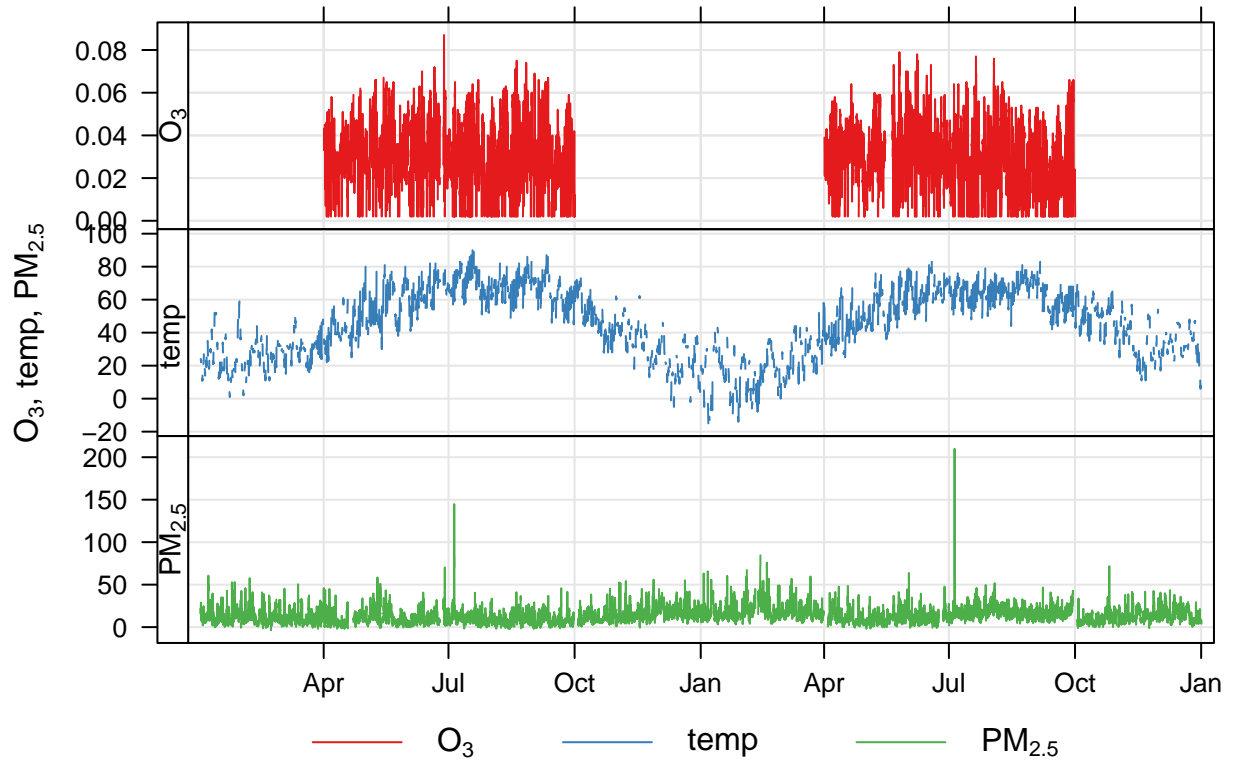
# reshape the data
site22_wide <- spread(site22, parameter, value)

# format the date column properly
site22_wide$datetime <- as.POSIXct(site22_wide$datetime, format = "%Y%m%dT%H%M",
                                   tz = "America/Chicago")

# some dates weren't converted--remove those
site22_wide <- filter(site22_wide, !is.na(datetime))

# we can't use rename() because the column names are numbers
# so we'll use names() <-
names(site22_wide) <- c("date", "ozone", "temp", "pm2.5")

timePlot(site22_wide, pollutant = c("ozone", "temp", "pm2.5"))
```



Solution 5

[Back to exercises](#)