**OPENAIR PRIMER**

Answer key for instructors and teaching assistants!

**EXPLORE OPENAIR**

1. (Slide 6 - *View Data*)  
   Once the data is imported, you can click on the object in your workspace to view it, or run the command head(data) to see some of the data.
   1. What do you notice? (Any interesting formatting of entries? Any columns you can recognize before going further?)

| <flexible and up to you> |
| --- |

* 1. (Slide 7 - *Columns*)  
     There are 13 variables in the dataset. Complete the following table with variable name to match the given meaning, format, and/or unit of measurement (done as a class)

| **Variable Name** | **Meaning** |
| --- | --- |
| site | The name of the collection location |
| code | The code for importing this site through the API |
| date | Date and time listed as yyyy-mm-dd hh:mm:ss |
| co | The level of carbon monoxide in μg/m^3 |
| nox | The level of other nitrous oxides (not no2 or no) in μg/m^3 |
| no2 | The level of nitrogen dioxide in μg/m^3 |
| no | The level of nitrogen monoxide in μg/m^3 |
| o3 | The level of ozone in μg/m^3 |
| so2 | The level of sulfur dioxide in μg/m^3 |
| pm10 | The level of particulate matter of 10 micron size in μg/m^3 |
| pm2.5 | The level of particulate matter of 2.5 micron size in μg/m^3 |
| ws | The speed of the wind in m/s |
| wd | The direction of the wind in compass degrees (0º north, clockwise around) |

1. (Slide 9 - *Plot 1: timePlot*)  
   The best way to do this clearly and concisely is to use the openair function timePlot() (plots the pollutant nox by default)  
     
   timePlot(data)  
     
   To specify one or more variables to plot, as well as the option to average over time:  
     
   timePlot(data,   
    pollutant = c("no", "no2", "nox", "o3"),   
    avg.time = "month")
   1. Do you see any gaps in this data? If so, why do you think they are there? If not, what would you do if there were gaps?

| <flexible and up to you> Gaps disappear with the month averaging, but a few are visible in the first plot, reasons might include that the sensor needs downtime for maintenance, etc. |
| --- |

* 1. Do you see any pattern of peaks or valleys in this data? Why do you think they are there?

| <flexible and up to you> You might comment on the fact that NOx and NO have their greatest peak at the same time in late 2000, and their peaks are generally correlated |
| --- |

* 1. You might notice that the time plots don’t have titles. What argument can you use from the R intro assignment to create a plot title? Try this out with an appropriate title of your choice, and submit your code

| timePlot(data,  pollutant = c("no", "no2", "nox", "o3"),  avg.time = "month",  main="Title")  <flexible and up to you> |
| --- |

1. (Slide 10 - *Plot 2: windRose*)  
   Run the command:  
     
   windRose(data)
   1. (Slide 11 - *windRose Questions*)  
      This plot produces 12 bars jutting out on a polar plot. What do the directions of these bars correspond to?

| From ? windRose: For windRose data are summarized by direction, typically by 45 or 30 (or 10) degrees and by different wind speed categories. |
| --- |

* 1. The legend below the plot lists a range of values for each color with units of (m s-1), what do the values correspond to (think about what variable has those units)?

| From ? windRose: Typically, wind speeds are represented by different width "paddles". The plots show the proportion (here represented as a percentage) of time that the wind is from a certain angle and wind speed range. By default windRose will plot a windRose in using "paddle" style segments and placing the scale key below the plot. |
| --- |

* 1. (Slide 12 - *windRose Questions*)  
     The plot is titled “Frequency of counts by wind direction (%).” If this is the case, what do the lengths of the bars represent (look at the units on the concentric rings)?

| From ? windRose: The plots show the proportion (here represented as a percentage) of time that the wind is from a certain angle and wind speed range. |
| --- |

* 1. In the lower right corner of the plot are values for the “mean” and “calm”, what do these mean (hint: mean is in m s-1 and you can check if the wind speed ever hits zero by making a plot of wind speed)?

| Mean: the mean wind speed in the data provided.  Calm: the percentage of given data when wind speed was calm or 0 m s-1 |
| --- |

* 1. (Slide 13 - *windRose Questions*)  
     Finally, using the above information as necessary, explain what a windRose shows (1-2 sentences).

| From ? windRose:  The traditional wind rose plot that plots wind speed and wind direction by different intervals. The pollution rose applies the same plot structure but substitutes other measurements, most commonly a pollutant time series, for wind speed. |
| --- |

* 1. (Slide 14 - *windRose Arguments*)  
     As before, we can use internal arguments to specify further.  
       
     windRose(data, type = "year", layout = c(3, 2))  
       
     How does type = “year” separate the data?

| It splits the data by year and produces a plot for each split. |
| --- |

* 1. How about type = “pm10”?  
       
     windRose(data, type = "pm10", layout = c(3, 2))  
       
     What is the data binned by now?

| From ? windRose: If that variable is numeric, then the data will be split into four quantiles (if possible) and labeled accordingly. If type is an existing character or factor variable, then those categories/levels will be used directly. |
| --- |

* 1. Explore the different variations of windroses given type and variable. Paste one that you produced of your choice, something that piques your interest. Explain what your windRose shows. Are there any conclusions you can draw?

| <flexible and up to you> |
| --- |

1. (Slide 15 - *pollutionRose*)  
   We can also make a variation of this called a pollution rose (nox by default):   
     
   pollutionRose(data)  
     
   To focus on a specific pollutant, use the pollutant argument:  
     
   pollutionRose(data, pollutant = "no")  
     
   To link with another pollutant, use the type argument:  
     
   pollutionRose(data, pollutant = "no", type = "no2", layout = c(4, 1))  
     
   To segment (remove the spaces between the bars) and normalize:   
     
   pollutionRose(data, pollutant = "nox", seg = 1, normalise = TRUE)
   1. The 12 bars correspond to the same data aspect as in the windRose, but now what does the shading correspond to?

| From ? pollutionRose (same help page as windRose): pollutionRose is a windRose wrapper which brings a pollutant forward in the argument list, and attempts to sensibly rescale breakpoints based on the pollutant data range by by-passing ws.int. |
| --- |

* 1. What does setting the segment to 1 and turning on normalization help you to see? (Hint: are the sizes of the colored regions relative to something? Also, think about how a pollution rose was employed by the article you read in assignment 2)

| From ? pollutionRose: seg: For pollutionRose seg determines the width of the segments. For example, seg = 0.5 will produce segments 0.5 \* angle.  normalise: If TRUE each wind direction segment of a pollution rose is normalized to equal one. This is useful for showing how the concentrations (or other parameters) contribute to each wind sector when the proportion of time the wind is from that direction is low. |
| --- |

* 1. This is the second time you’ve seen the layout argument, what does the argument accept and what does this do?

| From layout: ‘layout’ divides the device up into as many rows and columns as there are in matrix ‘mat’, with the column-widths and the row-heights specified in the respective arguments. |
| --- |

1. (Slide 16 - *pollutionRose challenge*)  
   Using the command pollutionRose() and knowledge of R and *openair* syntax, make a pollution rose for only the year 2002 that is tracking the level of pm10.  
     
   Submit your code you used to create this plot, including the commands to set up the file and import the data.

| library(openair)  data <- importAURN(site = "my1", year = 2003)  pollutionRose(data, pollutant="pm10")  OR  library(openair)  data <- importAURN(site = "my1", year = 2000:2005)  pollutionRose(selectByDate(data, year = 2003), pollutant="pm10") |
| --- |

1. (Slide 17 - *Plot 4: calendarPlot*)  
   calendarPlot(data)  
   calendarPlot(data, annotate = "ws")
2. Now, using your knowledge of R and openair, make a calendar plot for just the year 2003 tracking nitrous oxide only. Paste your code (script/data setup not necessary) and your plot below

| data2 <- importAURN(site = "my1", year = 2003)  pollutionRose(data2, pollutant="pm10")  OR  pollutionRose(selectByDate(data, year = 2003), pollutant="pm10") |
| --- |

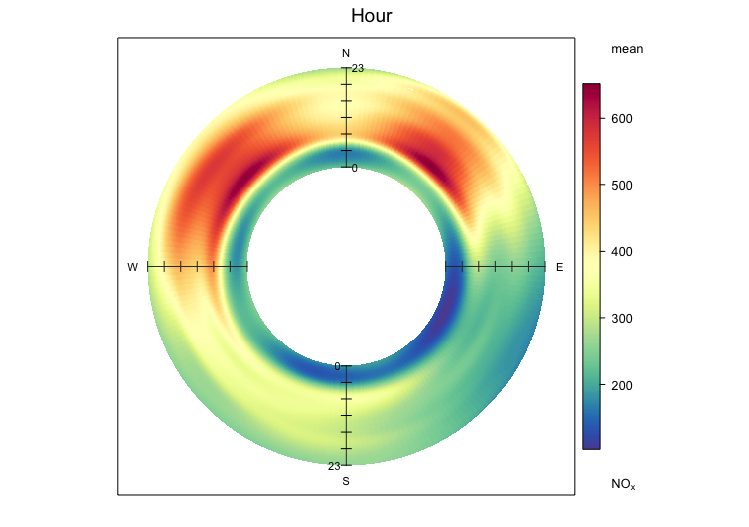
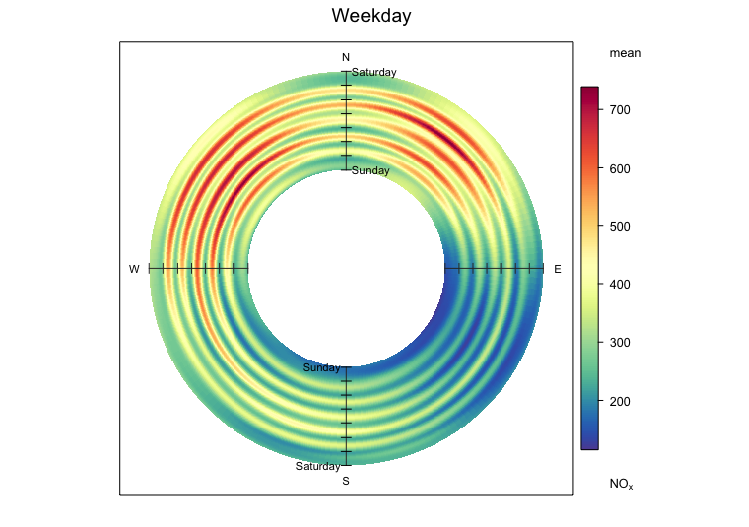
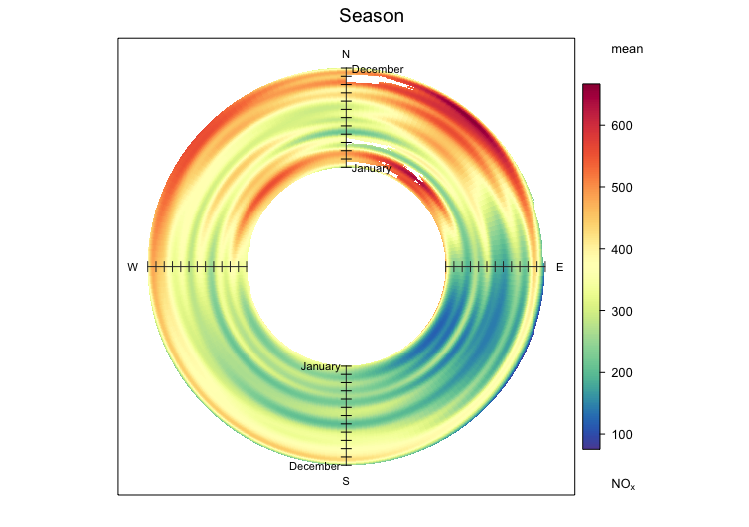
1. Imagine you made a timePlot with a monthly average of co for the whole time period (you can even do this now if you wish). Why might the calendarPlot be a useful next step?

| <flexible and up to you> finding a specific calendar date may be difficult on a time scale, but is quite clear on a calendar. |
| --- |

1. (Slide 18 - *Plot 5: polarAnnulus*)  
   Run each command, one at a time:

polarAnnulus(data, poll = "nox", period = "season", main = "Season")

polarAnnulus(data, poll = "nox", period = "weekday", main = "Weekday")

polarAnnulus(data, poll = "nox", period = "hour", main = "Hour"

1. First, how are these three most obviously different (think about what different arguments they have)?

| Different temporal periods to consider. |
| --- |

1. What do the colors correspond to? What does the location of that color correspond to?

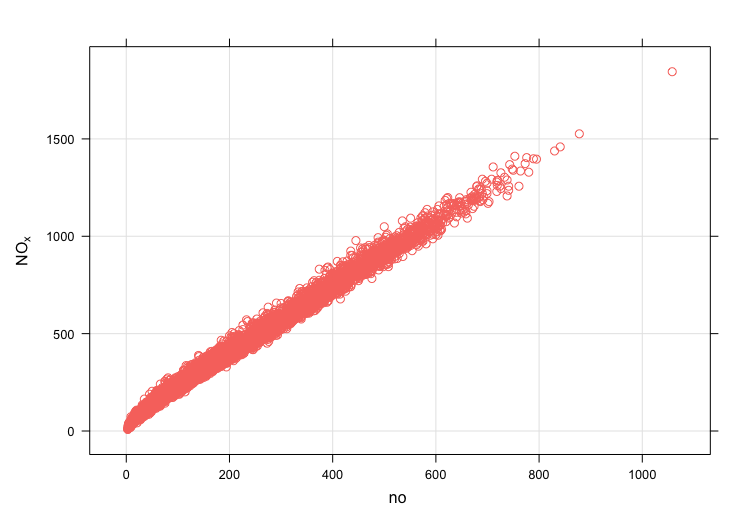
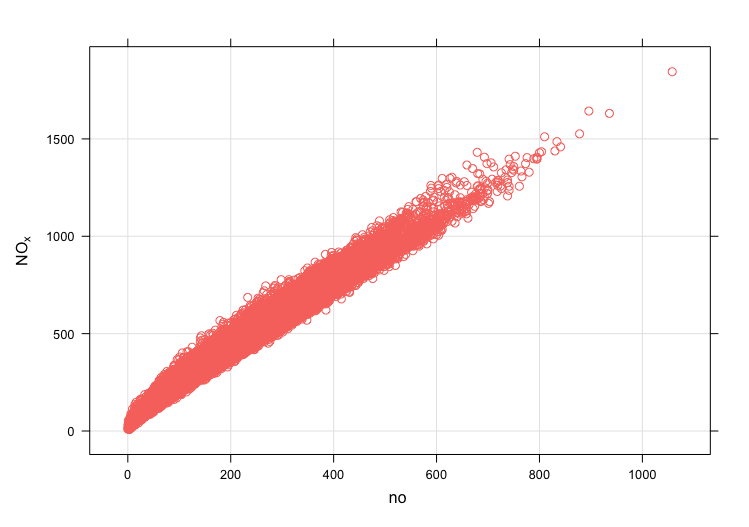
| From ? polarAnnulus: Typically plots the concentration of a pollutant by wind direction and as a function of time as an annulus. The function is good for visualizing how concentrations of pollutants vary by wind direction and a time period e.g. by month, day of week. |
| --- |

1. Why do we need to make a ring (an annulus) rather than just a full circle (what information might be lost as we approach the center of the circle and the circumference decreases)?

| From ? polarAnnulus:  Plotting as an annulus helps to reduce compression of information towards the center of the plot. |
| --- |

1. (Slide 20 - *Plot 6: scatterPlot*)  
   Finally, we’ll create a common plot for any science, the scatter plot. This command exists in base R, but openair has embellished it for our use.  
     
   Use the command scatterPlot to compare “no” on the x axis and “nox” on the y axis.

scatterPlot(data, x = "...", y = "...")

data2000 <- selectByDate(data, year = 2000)  


(Slide 21 - *scatterPlot (cont)*)  
Add to your scatterPlot command the arguments method="hexbin" and col="jet" to do this.

1. How did “hexbin” change the scatter plot? How is this helpful?

| From ? scatterPlot:  “hexbin” (hexagonal binning using the hexbin package) <flexible and up to you> |
| --- |

1. Experiment with other color scales (try help(openColours) and see the available “schemes”). Which scale is your favorite for a heatmap?

| <flexible and up to you> |
| --- |

1. Do you see any trend from the scatter plot? If so, what? You might consider if the variables have a linear relationship, if that relationship is positive or negative, etc.

| <flexible and up to you> Highly correlated! NO almost tracks 1:1 with NOx |
| --- |

1. (Slide 23 - *Closing Exercise: Your Choice!*)  
   As a closing exercise, go to the openair manual and find one plot that we **have not** seen or used today. Run the command to produce that plot for your data, paste it, and describe what the plot shows. The link to the manual is [here](https://bookdown.org/david_carslaw/openair/), and remember that all openair functions use similar internal arguments.

| <flexible and up to you> |
| --- |