Air Toxics Data in R

Kali Frost

October 27, 2015

Who we are

- Kali Frost
  - Research Associate at IU School of Public Health
- Nathan Byers
  - Database Analyst at Indiana University (IU) School of Medicine
- Eric Bailey
  - Web Application Developer at IU School of Medicine
- Former employees at the Indiana Department of Environmental Management, Office of Air Quality

Training Outline

- R basics
- Useful packages for air toxics
  - `raqdm`
  - `rucl`
  - `openair`
- Interactive web apps with `shiny`
- Yesterday's training covered 'zero to Shiny'
- Today, in the interest of time, this talk will assume some familiarity with R.
- If you are unfamiliar with R, check out some online resources provided later in these slides or you can view our training session yesterday. (`https://ebailey78.shinyapps.io/epaToxicsPresentation`)

The R Project

- R is free and open-source software for statistical computing and graphics
- Download here
- Statisitcal software that can do complex analyses using built-in functions, similar to SAS
- Also a programming language that is extendable (i.e. you can write your own functions/software)

R Basics - Command line

- In this presentation, command line code will be shown in blocks with gray background shading
- The output will be shown in a separate block below the command line input with a lighter shadow and ## at the beginning of each line
- Just need to remember:
  - Dark gray = User supplied commands
  - Light gray = R output

```
"command line"
```

```
## [1] "command line"
```

```
1 + 1
```

```
## [1] 2
```

R Basics - Resources

- Many great resources for learning R
- Beginners material
  - Quick R
  - UCLA
  - DataCamp
  - Code School
- Intermediate/advanced material
  - Cookbook for R
  - Coursera
  - Advanced R

Training Outline

- R basics
- Useful packages for air toxics
  - raqdm
  - rucl
  - openair
- Interactive web apps with shiny

Air Toxics related R Packages

raqdm - Overview

- Provides convenient access to US EPA's AQS Data Mart in R
- Makes use of the API discussed at `http://www3.epa.gov/airdata/toc.html`
- Additional details and sourcecode on github (`https://github.com/ebailey78/raqdm`)
- Still under development - Comments and Suggestions Welcome (`eb11307@gmail.com`)

raqdm - Features

- Query AQS Data Mart from the R console or through a convienient GUI
- Save your most common parameter options so you don't have to enter them repeatedly
- Access to all options available in EPA's web interface
- Import requested data directly into an R data.frame for further analysis
- Can do both synchronous and asynchronous data pulls from the AQS Data Mart

raqdm - Installation

- To install raqdm you will need the devtools package available on CRAN:

```
install.packages("devtools")
library(devtools)
install_github("ebailey78/raqdm")
library(raqdm)
```

raqdm - Setup

- You will need a username and password from EPA to access the data
- Request username and password from EPA by emailing aqsdatamart@epa.gov
- Once you have user credentials you can save them in raqdm with the setAQDMuser function

```
setAQDMuser("me@mystate.gov", "my_password", save = TRUE)
```

- You can set other default parameters with the setAQDMdefaults function

```
setAQDMdefaults(state = "18", bdate = "20140101",
                edate = "20141231")
```

- Setting save = TRUE will cause defaults to be saved across R sessions

raqdm - Requesting Data

- We've set defaults for Indiana (18), and bdate(20140101) and edate(20141231) in the previous slide
- Now we can request 2014 benzene data from Indiana with

```
benz_req <- getAQDMdata(param = "45201",
                        synchronous = FALSE)
# Request benzene
```

- Or we can request all available met data with

```
met_req <- getAQDMdata(pc = "MET", synchronous = FALSE)
```

raqdm - Retrieving Data

- Once the requests are processed on the server, we read in the data.

```
benz <- getAQDMrequest(benz_req)
met <- getAQDMrequest(met_req)
```

Exposure Estimates with `rucl`

rucl Overview

- rucl is an R package that assists in calculating Upper Confidence Limits of the Mean (UCLs) in R
- Based heavily on U.S. EPA's ProUCL software version 4.1. (http://www2.epa.gov/land-research/proucl-software)
- Needs additional development and testing before official release
- Not well documented yet

`rucl` Features

- Test for normal, lognormal, or gamma distributions
- Handles censored and uncensored datasets
- Can calculate over 30 different UCLs
- Will recommend UCLs based on data characteristics (experimental)

rucl Installation

- To install rucl you will need the devtools package available on CRAN:

```
library(devtools)
install_github("ebailey78/rucl")
library(rucl)
```

rucl Usage

- The primary function in rucl is ucl()
- The only required argument is a numeric vector that represents concentrations

```
# Create a random dataset with 50 values
# from a gamma distribution
x <- rgamma(50, 5, 2)
head(x)
```

```
## [1] 2.889196 2.647222 3.003525 1.988472 2.227599 3.52196
```

```
# Return the recommended UCL
ucl(x)
```

```
##   n.tucl   n.modt
## 2.760701 2.760745
```

rucl Usage

- You can also request specific UCL calculations

```
# Calculate the modified-t-based UCL for
# normal distributions
ucl(x, "n.modt")
```

```
##    n.modt
## 2.760745
```

rucl Usage

- Arguments can also be used to change the confidence level and number of bootstrap iterations

```
ucl(x, confidence = 0.95, N = 10000)
```

```
##    n.tucl    n.modt
## 2.760701 2.760745
```

```
ucl(x, confidence = 0.9, N = 10000)
```

```
##    n.tucl    n.modt
## 2.705258 2.705301
```

rucl Usage

- `type = detailed` will create a `rucl` object that contains a great deal of information about the dataset

```
ucl(x, type = "detailed")
```

`rucl` Usage

rucl Censored Data

- rucl can handle censored datasets using Kaplan-Meier
- Requires a second vector with TRUE/FALSE indicating whether the corresponding reading is a detect (TRUE) or nondetect (FALSE)
- For non-detects, assumes reported reading is detection limit

```
# Create a lognormal dataset with
# 50 readings censored at 0.4
x <- rlnorm(50)
d <- x > 0.4
x[!d] <- 0.4

ucl(x, d = d)
```

rucl Censored Data

```
## o.km.bcaboot
##     5.148966
```

Data visualization with `openair`

The openair project

- The `openair` package was created by Dr. David Carslaw at King's College London.
- The goals of the openair project are to create tools in R that use the wealth of air pollution data that is publicly available to make it easy to carry out sophisticated analyses quickly and in a reproducible way.
- Comprehensive user manual can be found here: (`http://www.openair-project.org/downloads/openairmanual.aspx`)

Importing data into openair

- We will use the `import()` function in openair to bring in a csv file that has been formatted for openair (we showed how to make this yesterday)
- The `import` funtion is helpful because it takes care of all of the final date formatting for openair plots

```
library(openair)
gary <- import(file = "Gary_openair.csv"), sep=",",
date="Date.GMT", time="X24.Hour.GMT",
date.format="%Y-%m-%d", time.format="%H:%M",
ws="Wind Speed - Resultant",
wd="Wind Direction - Resultant")
```

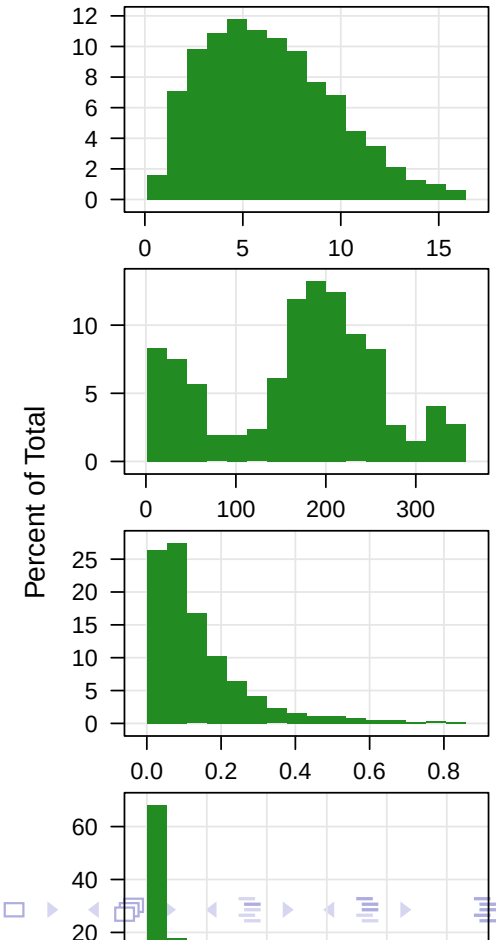
Summary plot

- Once the data is imported into `openair` we can use `summaryPlot` to quickly view the data
- Creates time-series plots and makes it easy to see chunks of missing data.
- The summary plot also displays basic summary stats such as mean, median and the 95th percentile.
- A histogram helps the user view the distributions of each of their parameters.
- You can call this plot using the following code:

```
summaryPlot(selectByDate(gary, year=2014))
```

Summary Plot

```
##         date1         date2 X24.Hour.GMT      Benzene
##     "POSIXct"     "POSIXt"     "factor"     "numeric"
##            wd            ws
##     "numeric"     "numeric"
```

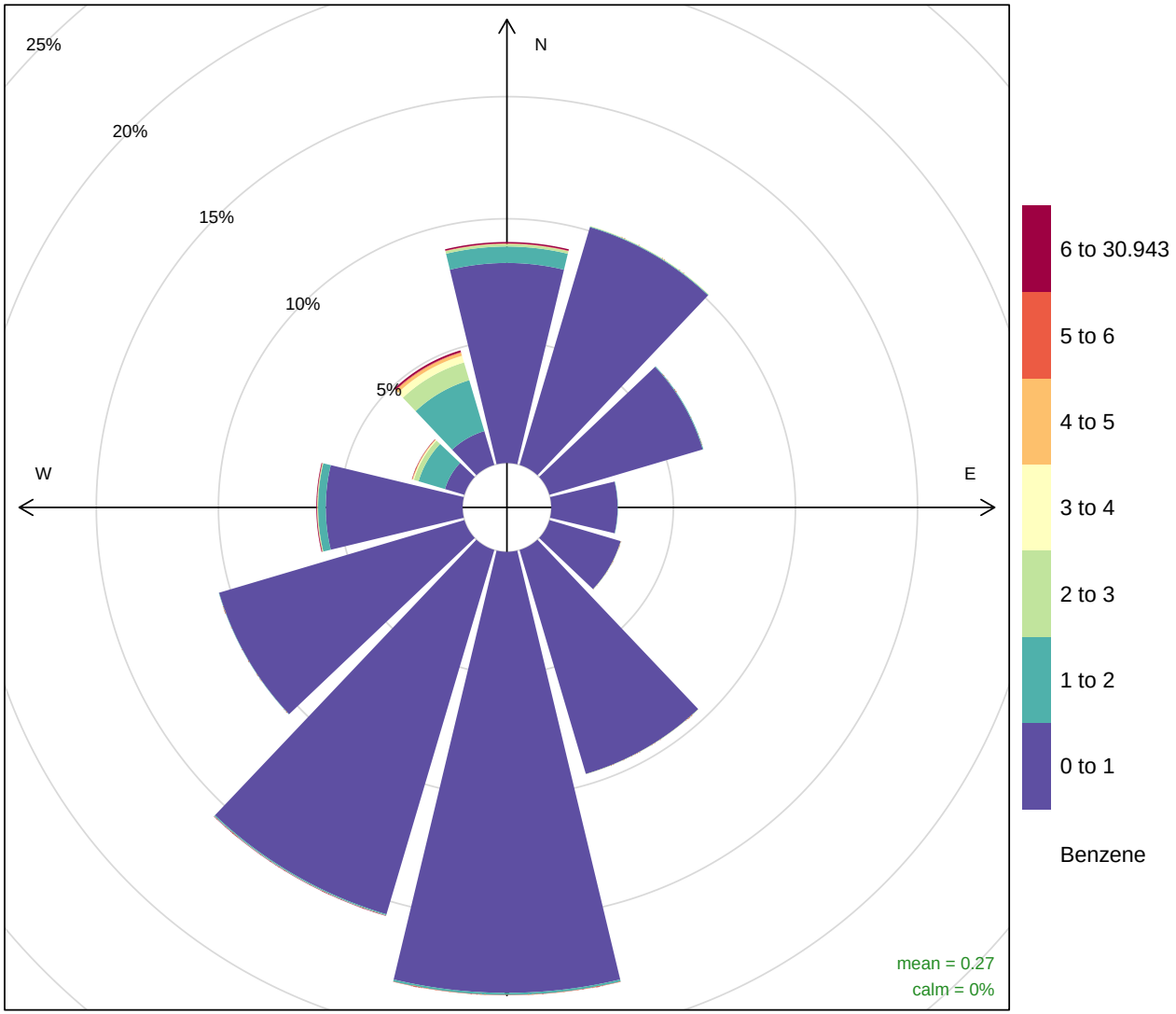


Pollution Rose

- A pollution rose plot is useful for describing the proportion of the contaminant that comes from each wind direction.

Pollution Rose

```
pollutionRose(gary, pollutant= "Benzene")
```

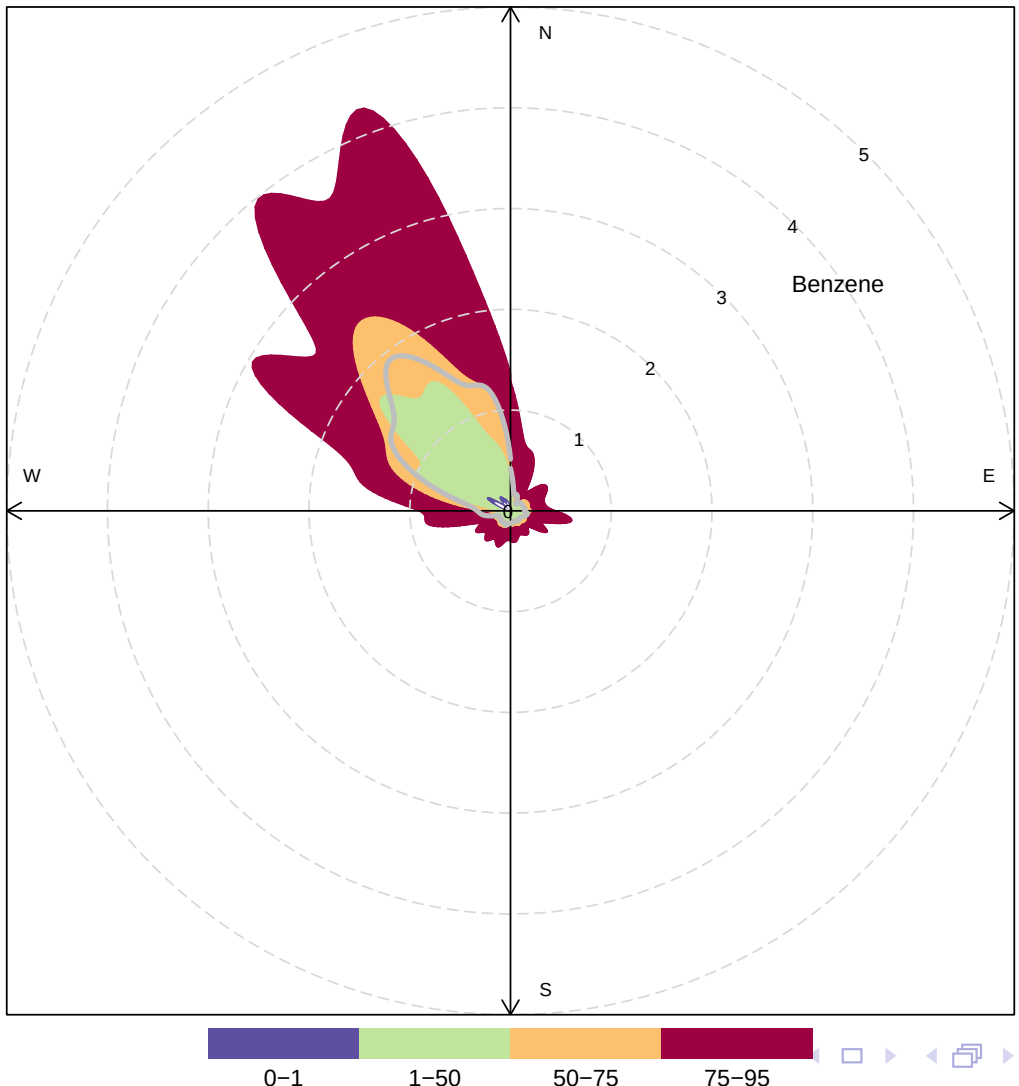


Percentile Rose

- A percentile rose plot can help you see the distribution of concentrations by wind direction

Percentile Rose

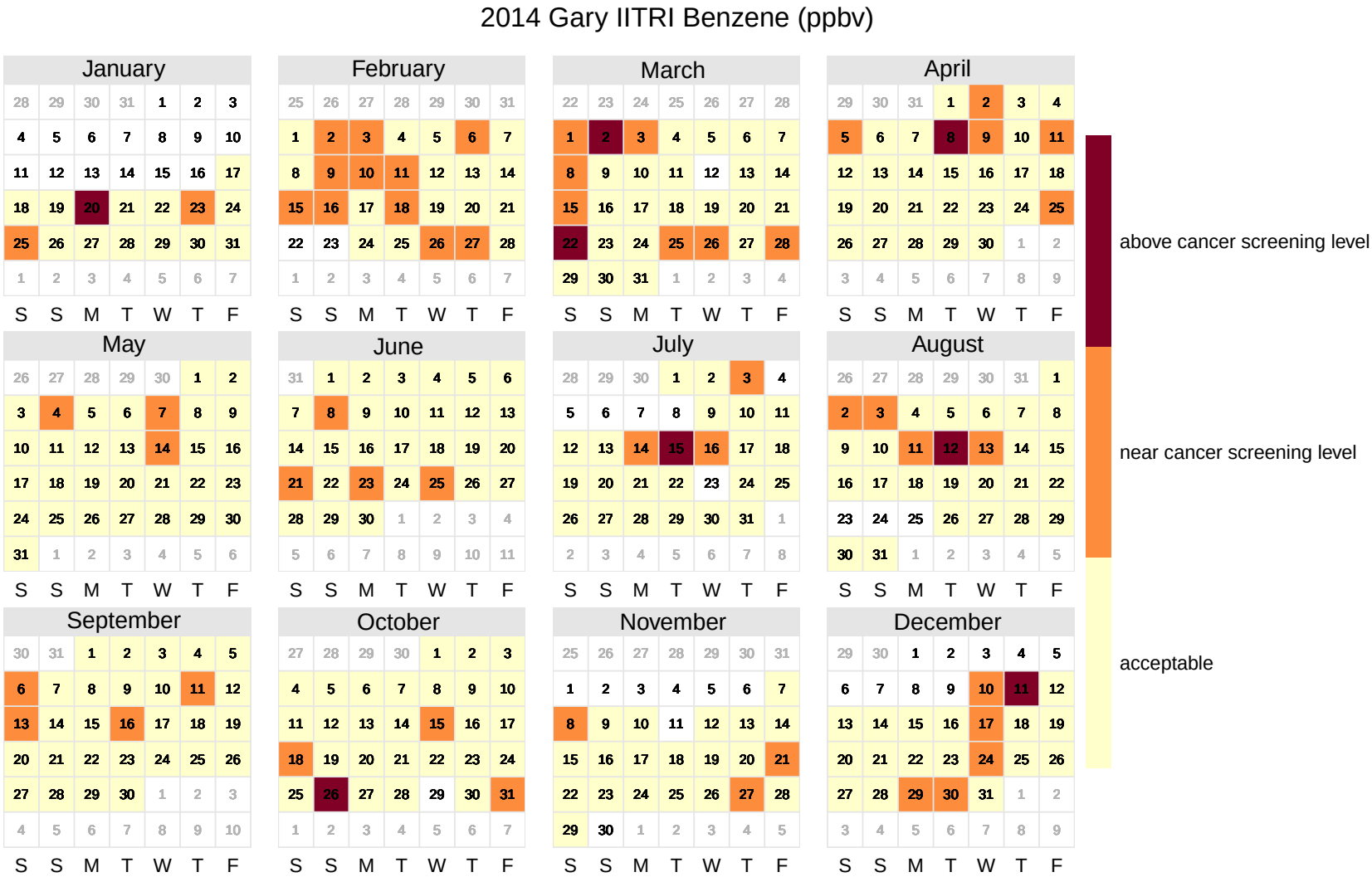
```
percentileRose(gary, pollutant="Benzene",
    percentile = c(0,1,50,75,95), smooth=TRUE)
```



Calendar Plots

```
calendarPlot(gary, pollutant = "Benzene",
  year=2014, statistic="mean",
  labels=c("acceptable", "near cancer screening level",
           "above cancer screening level"),
  breaks=c(0,0.4,1.4,200),
  main = "2014 Gary IITRI Benzene (ppbv)")
```

Calendar Plots



Training Outline

- What is R and RStudio
- R basics
- Useful packages for air toxics
  - raqdm
  - rucl
  - openair
- Interactive web apps with `shiny`

Shiny - Overview

- `shiny` is a package developed by RStudio
- Makes it easy to create web applications using the R language
- Not necessary to know HTML, CSS, or JavaScript
- Let's look at some apps we've created

ToxWatch

- **Input**
  - 15 years of sample data
  - 1-in-6 day sampling (TO-14 and TO-15)
  - 9-12 monitors at any given time
  - 62 pollutants
  - Stored in Oracle database
  - User selects any combination of pollutants/monitors/dates
- **Output**
  - Raw data (also as csv download)
  - Risk/Hazard Estimates
  - Time Series
  - Boxplots

Decisions

- When the ToxWatch app was being created there seemed to be a real shift towards "using the noise" in calculating exposure estimates.
- Use whatever value the GC/MS returned even if it was below the MDL and treat NDs as 0.
- Not necessarily recommending this, but easier to calculate UCLs.
- Sometimes analysts don't have ready access to all of the MDL information required for using K-M or ROS.
- In an ideal world, we would incorporate the `rucl` package functionality into the app
- The app, as it is now, is calculating a student's-t UCL
- "ToxWatch App" - `https://stats.idem.in.gov/toxics/`

Indiana Metals Data

- 2000 - 2013 1-in-6 day metals and PM data
  - TSP, PM 10, PM 2.5,
  - As, Be, Cd, Cr, Pb, Mn, Ni
- Incorporates EPA Xact Monitor Data for 2012-2013 near US Steel facility
- Emissions bubble maps for 2002, 2005, 2008 and 2011 NEI metals data
- Warning: Data has not been processed very much, for demonstration only
- "Indiana Metals Data" - `https://kfrost.shinyapps.io/INmetals/`

SO2 Modeling Study

- The Gibson Spatial Analysis (GSA) tool is a web application that takes output from AERMOD modeling runs and helps the user visualize model performance spatially
- More specifically, this tool describes the modeling study conducted by IDEM at the Gibson Generating Station in Southwest Indiana
  - 1-hour SO2 modeling analysis using AERMOD v12345
  - Hourly SO2 monitoring data was collected at four monitors (Mt. Carmel, Coal Rd, East, and Schrodt) near the source and used for comparison with the model
  - Good spatial coverage provided by four monitors provided hourly SO2 background needed to compare model and monitor results
- "Gibson Spatial Analysis Tool"
  -`https://stats.idem.in.gov/GSA-new/`

Contact Information

- Nathan Byers, natebyers@gmail.com
- Kali Frost, kdfrost14@gmail.com
- Eric Bailey, eb11307@gmail.com