

# Package ‘aiRpollution’

September 22, 2015

**Type** Package  
**Title** CCAAPS air pollution exposure predictions  
**Version** 0.1  
**Date** 2015-09-08  
**Maintainer** Cole Brokamp <cole.brokamp@gmail.com>  
**Description** Produce elemental PM concentration predictions for specific locations based on the Cincinnati Children's Asthma and Air Pollution Study (CCAAPS)  
**Depends** R (>= 3.1.2)  
**Imports** CB,  
rgdal,  
rgeos,  
sp,  
raster,  
randomForest,  
stringr,  
pbapply  
**License** GPL  
**LazyData** TRUE

## R topics documented:

depIndex . . . . .	1
distanceToClosest . . . . .	2
greenspace_static . . . . .	3
linesLength . . . . .	3
predictPollution . . . . .	4
<b>Index</b>	<b>5</b>

---

depIndex	<i>Extract the deprivation index</i>
----------	--------------------------------------

---

**Description**  
Deprivation index derived from PC analysis of 8 different SES Census Tract variables from the 2010 ACS.

**Usage**

```
depIndex(loc, buffer.radius = 0)
```

**Arguments**

<code>loc</code>	loc spatial object (with valid proj4string)
<code>buffer.radius</code>	if set to 0, returns the dep index of the containing census tract, if set to > 0, returns the mean dep index for the census tracts that are at least partially contained in the buffer circle

**Value**

data.frame, named `dep.index` or `dep.index_buffer.radius`

---

<code>distanceToClosest</code>	<i>Distance to closest</i>
--------------------------------	----------------------------

---

**Description**

Function which calculates the distance to a closest spatial object. Anything that works with `rgeos::gDistance` will work in this function. `loc` will be reprojected to the projection of the `lines.shapefile` and the units will be of that projection.

**Usage**

```
distanceToClosest(loc, lines.shapefile)
```

**Arguments**

<code>loc</code>	spatial object (with valid proj4string)
<code>lines.shapefile</code>	spatial object for which to calculate minimum distance

**Value**

data.frame with distance; named based on `lines.shapefile` input (units are assumed to be meters; if feet back transform output by 0.3048006096 m/ft)

---

greenspace_static	<i>Calculate greenspace</i>
-------------------	-----------------------------

---

**Description**

Uses the greenspace raster (ndvi2000\_3735.tif) to extract average greenspace values

**Usage**

```
greenspace_static(loc, buffer.radius = 400)
```

**Arguments**

loc	spatial object (with valid proj4string)
buffer.radius	(radius of circle in which to calculate average NDVI)

**Value**

data.frame with value; named greenspace\_buffer.radius

---

linesLength	<i>calculate the length of lines within a buffer radius</i>
-------------	---

---

**Description**

calculate the length of lines within a buffer radius

**Usage**

```
linesLength(loc, lines.shapefile, buffer.radius = 100)
```

**Arguments**

loc	sp object with coordinates and valid proj4string
lines.shapefile	shapefile of lines to use in calculation
buffer.radius	buffer radius in meters (assumes that lines.shapefile is projected to unit of feet)

**Value**

data.frame named based on buffer radius and lines shapefile

---

predictPollution	<i>predict pollution exposure</i>
------------------	-----------------------------------

---

### Description

This function is an implementation of the elemental PM exposure (random forest and regression) models developed for the Cincinnati Children's Asthma and Air Pollution Study (CCAAPS). The underlying functions are not available to the user. Not meant to be a generalizable package, its sole purpose is to generate exposure estimates for the Cincinnati area. It will return NA if any of the necessary predictors are not available for the location.

### Usage

```
predictPollution(loc, element, model.type, prog.bar = TRUE)
```

### Arguments

loc	the location for which to estimate the concentration (must be a spatial object and have a valid proj4string)
element	element for which to predict the concentration (one of "Cu", "Fe", "Zn", "S", "Ni", "V", "Si", "K", "Pb", "Mn", "Al", "TRAP", "PM25")
model.type	either "rf" for random forest or "lm" for regression
prog.bar	logical, show a progress bar?

### Examples

```
library(sp)
sample.loc <- data.frame('x'=-84.5371597, 'y'=39.1603015)
coordinates(sample.loc) <- c('x', 'y')
proj4string(sample.loc) <- CRS("+init=epsg:4326")
predictPollution(loc=sample.loc, element='TRAP', model.type='rf')
```

# Index

depIndex, [1](#)  
distanceToClosest, [2](#)  
greenspace\_static, [3](#)  
linesLength, [3](#)  
predictPollution, [4](#)