Examining the effect and its temporal variations of near-road greenspaces for different types of roads on regional PM2.5 levels – a case study in London

(The Independent, 2014)

Yulun Lin (19040824)

2022

This dissertation is submitted as an Independent Geographical Study as a part of a BSc degree in Geography at King's College London.

KING'S COLLEGE LONDON

UNIVERSITY OF LONDON

DEPARTMENT OF GEOGRAPHY

INDEPENDENT GEOGRAPHICAL STUDY

I, Yulun Lin, hereby declare (a) that this dissertation is my own original work and that all source material used is acknowledged therein; (b) that it has been specially prepared for a degree of King's College London; and (c) that it does not contain any material that has been or will be submitted to the Examiners of this or any other university, or any material that has been or will be submitted for any other examination.

This Dissertation is 8134 words.

Signed: Yulun Lin

Date: 7th April 2022

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. James Millington, without whom this Independent Geographical Study would have never become possible.

I would also like to thank my family and friends for the support and company during the completion of this IGS as well as throughout the three years of my undergraduate course.

ABSTRACT

The effect of near-road green space on regional $PM_{2.5}$ levels was examined using data from 21 London air quality monitoring sites. A linear regression model was used to investigate the relationship between the annual mean regional $PM_{2.5}$ concentrations and the conditions of green spaces near different classes of roads. The temporal variations in the relationship were also explored by modelling the regional average $PM_{2.5}$ concentrations over different periods as functions of the conditions of green spaces near different types of roads. The results show that near-road green spaces for Unclassified and Other roads have a negative effect on regional $PM_{2.5}$ levels, while the other types have a positive effect, and there are large temporal variations detected. The findings contribute to the planning and application of near-road green spaces in terms of PM pollution mitigation.

Key words: fine particulate matter; near-road greenspace; multivariate linear regression model

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Particulate matter (PM) refers to small solid and liquid matter suspended in the air, and is one of the most serious threats to human health among all the ambient air pollution. High exposure to PM can cause damage to the human body including the lung (Löndahl *et al.*, 2006), the heart (Sun *et al.*, 2010; Brook *et al.*, 2010) and the airway (González-Flecha, 2004), depending on the size of the particle. PM is mainly classified into two categories according to their aerodynamic diameter (Kim *et al.*, 2015), namely fine particulate matter ($PM_{2.5}$), which has a diameter smaller than 2.5 μm, and coarse particulate matter (PM10), which has a diameter between 2.5 to 10 μm. The sizes of PMs decide their transport abilities in the atmosphere as well as in the human body. $PM_{2.5}$ tends to travel longer in the atmosphere and penetrate deeper into the human body than PM10. As a result, major health problems related to $PM_{2.5}$ are associated with the lungs, Bronchi branches and Bronchioli (Löndahl *et al.*, 2006), while PM10 mainly causes damage to respiratory systems (airway) (González-Flecha, 2004). It is estimated that more than two million deaths worldwide each year are directly related to diseases caused by air pollution, most of which by fine particulate matter (Shah *et al.*, 2013). $PM_{2.5}$ is the primary contributor to human health issues relating to ambient air pollution.

The hazard mainly comes from exposure to a high concentration of $PM_{2.5}$, and the lower the concentration, the less the danger it exposes to human health. The WHO guideline value for $PM_{2.5}$ is 15 μg/m3 daily mean or 5 μg/m3 annual mean (WHO, 2021). This guideline represents the highest possible concentration to which the effect of $PM_{2.5}$ on human health is acceptable, but does not guarantee no damage to health. However, most regions around the world, especially regions in developing countries, have $PM_{2.5}$ levels higher than the WHO guidelines (World Bank, 2017).

The fine particulate matter in the atmosphere comes from both anthropogenic and natural sources. The former include combustion of fossil fuels, industrial and agricultural activities, and erosion of pavement by road traffic

(Srimuruganandam and Nagendra, 2012). The natural sources include volcanoes, wildfires, dust storms and sea spray (Anderson *et al.*, 2012). Natural sources contribute only 18% to global $PM_{2.5}$ pollution, with the rest from anthropogenic sources, among which traffic section takes up the highest percentage (Karagulian *et al.*, 2015). Hence, finding a way to mitigate the $PM_{2.5}$ pollution from road transport emissions can greatly reduce the $PM_{2.5}$ level in urban areas.

One proposed approach to this is developing near-road greenspaces. Many researchers have examined the effect of green spaces in reducing regional PM levels (Kończak *et al.*, 2021; Song *et al.*, 2015; Nowak *et al.*, 2006; Lei *et al.*, 2018; Irga *et al.*, 2015; Beckett *et al.*, 2000; Hofman *et al.*, 2016), which is mainly through two mechanisms - mass removal and transmission block. On one hand, vegetation in green spaces can help directly remove the PM from the air by capturing and storing them on the leaf surface as well as in the wax layer (Kończak *et al.*, 2021). On the other hand, green spaces can act as a 'windbreak' that interrupts the dispersion of particulate matter (Morakinyo and Lam, 2016) as well as alter other local meteorological environments including temperature, barometric pressure, relative humidity, etc. which also affect PM level (Hofman *et al.*, 2016).

Based on these theoretical and empirical foundations, near-road greenspaces are believed to have a positive effect on lowering PM concentrations. Indeed, there have been studies finding near-road air quality is significantly improved by vegetation (Morakinyo and Lam, 2016), especially on busy roadsides in open areas (Baldauf *et al.*, 2011). However, Vos *et al.* (2013) found that in some cases, instead of reducing $PM_{2.5}$ concentration, roadside vegetation can actually enhance PM pollution nearby by hindering the wind flow and resulting in an accumulation of particulate matter in the area (Abhijith *et al.*, 2017). Such a finding brings uncertainty to the effect of near-road greenspaces on lowering $PM_{2.5}$ levels in urban areas, and further investigation is needed.

Previous researches on examining the effect of green spaces, especially near-road greenspaces, in reducing PM concentration in urban areas can be divided into two streams. The first stream primarily focused on assessing the abilities in capturing particles in the air of greenspaces. Liu *et al.* (2015) found that canopy density, leaf area, mean diameter at breast height, average tree height and grass coverage and height in forests could greatly alter $PM_{2.5}$ concentration. Jeanjean *et al.* (2017) and Steffens *et al.* (2012) suggested that vegetation has an overall higher ability in reducing PM pollution during summer because of higher leaf area density. Different vegetation species also have different levels of impact on PM concentrations. For example, cypress trees reduce PM levels more than pine trees (Ji and Zhao, 2014). Variations in the location of vegetation in relation to wind direction can also lead to changes in its ability to reduce PM concentrations (Al-Dabbous and Kumar, 2014). Greenspaces are most effective in reducing PM concentrations when the wind blows from areas of high PM levels (e.g. roads) towards them. Lei *et al.* (2018) found that patterns of greenspaces can also influence their ability to reduce PM pollution. Increasing the differences between areas of greenspace patches as well as their edge complexities can significantly lower PM concentrations. A series of meteorological factors including wind (Przybysz *et al.*, 2018; Wang *et al.*, 2015Przybysz *et al.*, 2014; He *et al.*, 2020), precipitation (Xu *et al.*, 2017; Wang *et al.*, 2015) and solar radiation (temperature) (Wang *et al.*, 2015) can also change the effect of green space on reducing $PM_{2.5}$ levels. These research findings contribute extensively to the academic understanding and policy-making of the urban greenspaces in tackling PM pollution. However, most of them failed to consider temporal changes. $PM_{2.5}$ concentrations in different seasons can vary greatly, and even within one day, the concentrations have highs and lows. In these cases, the influences of greenspaces on PM concentrations could also be changing. Moreover, the studies that did consider temporal changes all had an approach that was through field measurements, which, while delivering valuable first-hand data and solid mechanism-level understandings, were not convincing enough if were to be applied to a larger scale.

The other stream that includes this subject used land-use regression (LUR) extensively to examine how land use types affect spatial-temporal changes in

$PM_{2.5}$ levels. Wu *et al.* (2017) utilized a LUR model with $PM_{2.5}$ concentrations and monthly NDVI (Normalized Difference Vegetation Index) data in Taipei, and found a strong negative correlation between them. Xu *et al.* (2019) also found a relation between forest land type and $PM_{2.5}$ level through LUR. However, the problem with the LUR technique is that it always suffers from multicollinearity (Ross *et al.*, 2007), which makes the model output less reliable. To overcome this, Kim (2020) developed a partial least-squares regression model, which minimizes the influence of multicollinearity of the variables, to study the effects of land use on PM levels in different seasons in Seoul, South Korea, and found that the percentage of green space area is negatively related to regional PM concentrations. Yet, none of them was able to evaluate the relation between near-road greenspaces and regional $PM_{2.5}$ concentrations.

Therefore, although urban greenspaces have been proven to have a significant effect on PM reduction, the influence of near-road greenspaces and the temporal changes in the influence are still not clear. Given the fact that road traffic is the largest contributor to $PM_{2.5}$ pollution in most parts of the world (Karagulian *et al.*, 2015), it is important to determine whether near-road greenspaces have a positive or negative effect on reducing PM concentrations. Hence, this study aims to examine the role of near-road greenspaces with regard to $PM_{2.5}$ concentration, taking London as a case study city. To be more specific, $PM_{2.5}$ data from 21 selected air quality monitoring sites across London were used to examine the relationships between different types of near-road greenspaces and $PM_{2.5}$ concentration as well as the temporal changes in the relationships. It is recognised that greenspaces that are near different types of roads will have different effects on reducing PM concentrations, so the near-roads greenspaces were classified into several categories according to their road types. The result of this study can enrich the understanding of near-road greenspaces' effect on lowering regional $PM_{2.5}$ concentrations and its temporal change, and inform local urban green space planning.

2. Methods

2.1 Background

London has a population of approximately 9 million and covers a land area of around 1500 km2. It is the largest city in the UK and has one of the busiest road traffic in the country. It is characterised by a temperate oceanic climate, with warm to hot summer and cool winter, and high precipitation all year.

$PM_{2.5}$ pollution in London

The annual average $PM_{2.5}$ level in London was reported as 13.3 µg/m3 in 2016 (Mayor of London, 2019), which was above the WHO guideline for annual mean concentration (5 µg/m3). It is estimated that apart from transboundary sources, the largest proportion (30%) of the $PM_{2.5}$ pollution comes from the road transport section (Mayor of London, 2019). In areas with intensive traffic flow (e.g. central London), the $PM_{2.5}$ may be much higher than the annual average level.

Air Quality Monitoring in London

London has one of the largest air quality monitoring networks in the world, with participation from all kinds of organisations and departments. The LAQN (London Air Quality Network) is one of them and is operated by the Environmental Research Group at Imperial College London, in cooperation with TfL (Transport for London), Defra (Department for Environment, Food and Rural Affairs) and local authorities where the monitoring sites are located (London Air, 2022a). It provides the public with open air quality data collected from its monitoring sites all across London. Apart from LAQN, the AURN (Automatic Rural and Urban Network) is another network that provides nationwide hourly air quality data to the public (Defra, 2022), with several sites in London.

The richness of the air quality data is a very important reason for choosing London as the case study city. The spatial change of $PM_{2.5}$ concentrations across London is a very important dimension of this study, hence it is crucial to gather data from different monitoring sites.

Road classification

The roads in London (and the UK in general) are classified into four categories (GOV.UK, 2012):

1. A roads - major roads aiming to provide transport links within or between areas. This type of road should have the highest volume of traffic of the four.

2. B roads - a lower class of roads, often with a poorer physical standard. Intended to feed traffic between A roads and smaller roads

3. Classified unnumbered roads - smaller roads connecting A, B roads with unclassified roads. Also known as C roads

4. Unclassified roads - local roads supporting local traffic. Most roads in the UK fall into this category. This class of roads have the lowest volume of traffic.

Except for those four categories, the motorway is another category of road that provides high-speed long-distance transportation. The number of motorways is much lower than the other four types of roads.

Greenspace in London

London is a green city, with roughly 40% of its area being greenspaces. However, the greenspaces are not evenly distributed across the whole city, with a much larger portion in Outer London and a smaller portion in Inner London. The uneven spatial distribution pattern gives an opportunity to study its relationship with regional air quality, and in this case, with regional $PM_{2.5}$ levels.

There are currently two schemes to protect the urban greenspaces in London, with one focusing on protecting undeveloped land around the city called Green Belt and the other aiming to protect greenspaces within the city called Metropolitan Open Land (MOL). The two designations helped develop and maintain extensive urban green areas in London. 22% of London is specified as Green Belt and another 10% is specified as MOL (GiGL, 2018).

2.2 Data sources

$PM_{2.5}$ data

Hourly mean $PM_{2.5}$ concentration data from 21 air quality monitoring sites across London were downloaded from the London Air website (London Air, 2022b), which is the website of the LAQN. Most of the sites are in the LAQN and the others are in the AURN. The 21 selected sites are located mainly in Inner London, with 2 of them in Outer London. **Figure 1** shows their locations. Their location information was downloaded from London Datastore (London Datastore, 2019).

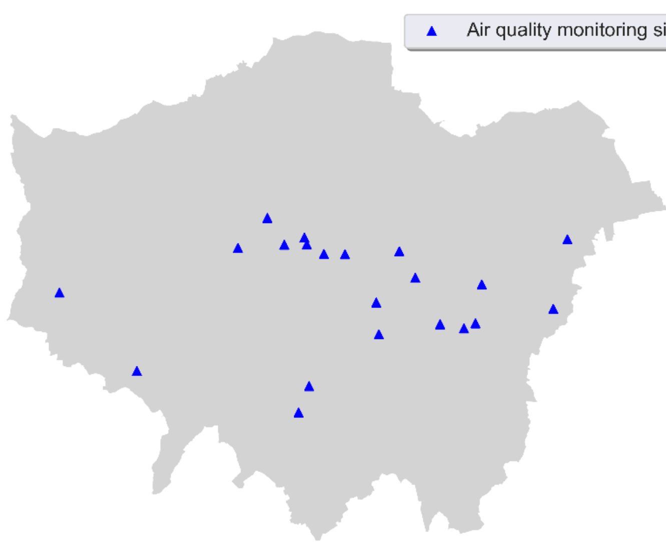


Figure 1: Locations of the 21 air quality monitoring sites in London.

The hourly mean $PM_{2.5}$ concentration data is for the year 2019. This is due to the concern about the impact of the COVID-19 pandemic and lockdown since March 2020. During the lockdown, most $PM_{2.5}$ sources (road traffic in particular) were significantly reduced (Wang and Li, 2021), and therefore the $PM_{2.5}$ level was much lower than normal (pre-COVID level). To minimize the interference, the study period was determined to be the most recent year prior to the pandemic.

It is noteworthy that the $PM_{2.5}$ data used are all provisional (not ratified), so the result of this study should be evaluated and used with caution.

Greenspace

The greenspace information was generated from the OS MasterMap Greenspace Layer (OS, 2021a) provided by the Ordnance Survey, which is the UK's national mapping agency. The MasterMap Greenspace Layer contains all accessible (public parks, sports facilities, etc.) and non-accessible (private garden) urban green spaces in the UK. The map is in the form of vector data divided into 5km x 5km grids. 26 grids were downloaded from the EDINA Digimap Ordnance Survey Collection (Digimap, 2021), which is a collection of OS data owned by EDINA at the University of Edinburgh. The MasterMap Greenspace Layer is updated twice a year, in April and October respectively. The dataset used in this study is from October 2019 in order to synchronise with the $PM_{2.5}$ data.

Road

The road information was generated from the OS Open Roads (OS, 2021b) which is also provided by the Ordnance Survey. This dataset contains not only the spatial geometry of every road in the UK, but also their information such as classification, name, function, etc. The road dataset is also in the form of vector data with a grid size of 100km x 100km, and is also updated twice a year in April and November. The data used in this study is from November 2019.

2.3 Data pre-processing and EDA

For the investigation of near-road greenspaces' effect on regional $PM_{2.5}$ concentrations, this study focus on examining the relationship between the $PM_{2.5}$ data from each air quality monitoring site and the near-road greenspace conditions in the 1km surrounding area around each site. The 1km buffer was decided based on some previous studies (Lei *et al.*, 2018; Chen *et al.*, 2019; Cai *et al.*, 2020) that investigated the effect of urban greenspaces on $PM_{2.5}$. Before the analysis, some data pre-processing procedures were performed, and explanatory data analysis was then conducted on both the dependent and independent variables.

Data cleaning for $PM_{2.5}$ data

Before the analysis, the $PM_{2.5}$ data was first cleaned. This includes (the concrete process in the Appendix):

1. removing unusual values - some of the values that were very abnormal (e.g. negative $PM_{2.5}$ readings) were removed (set to be null).

2. filling missing values - then all the null values were filled using a technique called the mean-before-after method (Norazian *et al.*, 2008), which is to replace a missing value with the mean of the data points before and after it. In cases where there were several continuous missing values, the closest non-null data points before and after the missing value were used to generate the replacement using linear regression. If the number of continuous missing values exceeded 12 (i.e. half a day), the 12 missing values were then replaced with the values from the same period of the previous day. This is a method commonly used in dealing with missing values within environmental datasets (Chen and Xiao, 2018).

After removing all unusual values and filling all missing values, an initial observation of the $PM_{2.5}$ data found an annual mean concentration of 11.8 μg/m3 with all 21 sites exceeded the WHO guideline of 5 μg/m3, as shown in **table 1**. However, compared to the reported 13.3 μg/m3 annual mean in 2016, most of the sites had a lower annual mean, which proved that London's past efforts on reducing $PM_{2.5}$ pollution have been working, although the pollution level is still significantly harmful to human.

Siteid	*Sitename*	*Annual mean $PM_{2.5}$ concentration (µg/m3)*
BX9	Bexley - Slade Green FDMS	11.2
BL0	Camden - Bloomsbury	10.9
CD9	Camden - Euston Road	13.7
CD1	Camden - Swiss Cottage	11.1
CT2	City of London - Farringdon Street	13.9
CT3	City of London - Sir John Cass School	12.1
CR8	Croydon - Norbury Manor	10.1
GR4	Greenwich - Eltham	10.9
GB0	Greenwich - Falconwood FDMS	12.6
GN6	Greenwich - John Harrison Way	11.0
GN3	Greenwich - Plumstead High Street	13.4
GR9	Greenwich - Westhorne Avenue	10.5
HV1	Havering - Rainham	11.4
LH0	Hillingdon - Harlington	9.4
KC1	Kensington and Chelsea - North Ken	9.6
HP1	Lewisham - Honor Oak Park	9.9
LW2	Lewisham - New Cross	15.4
TD5	London Teddington Bushy Park	11.7
ST5	Sutton - Beddington Lane north	11.7
TH4	Tower Hamlets - Blackwall	12.6
MY7	Westminster - Marylebone Road FDMS	14.2
-	**Annual mean for the Whole London**	**11.8**

Table 1: Annual mean $PM_{2.5}$ concentration for each site and the annual mean for London in 2019.

In terms of daily mean, London's $PM_{2.5}$ concentration exceeded the WHO guideline of 15 µg/m3 on 74 out of 365 days, as illustrated in **Figure 2**. Most of these days were in winter (November to January) and spring (February to April), which revealed a fluctuation in the annual trend with more days of high concentrations and higher monthly means during November to April and fewer days and lower monthly means during May to October. As other studies have shown (Lei *et al.*, 2018; Kim, 2020; Liu *et al.*, 2014; Li *et al.*, 2015), there is a very strong seasonal difference in the $PM_{2.5}$ level, and the presence of such difference could have an impact on the relationships between near-road greenspace and $PM_{2.5}$ concentration.

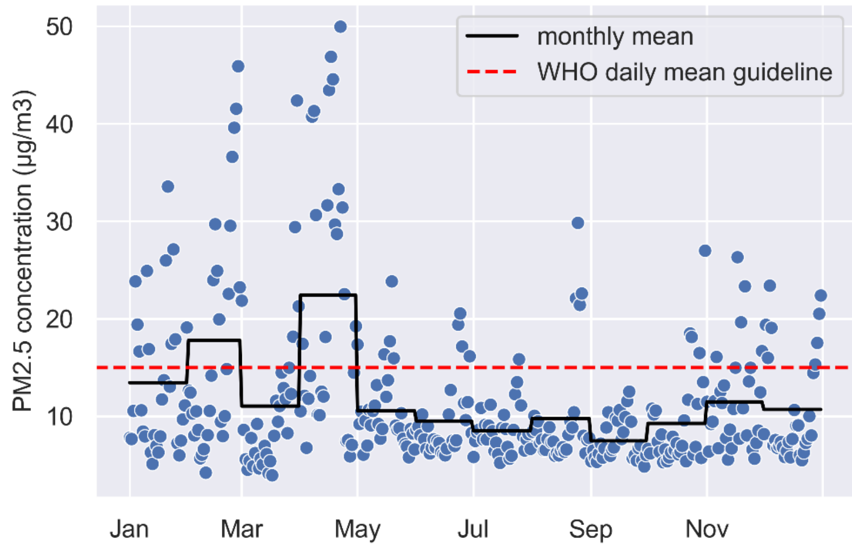


Figure 2: daily mean concentrations of $PM_{2.5}$ in 2019 compared to WHO guideline and monthly mean.

Similarly, the fluctuation in the daily trends of London's $PM_{2.5}$ level is also notable. As **Figure 3** shows, there are two peaks in $PM_{2.5}$ concentration throughout the day - one between 7-9 am with a concentration of around 12.5 µg/m3, and the other around midnight with a concentration over 13 µg/m3. The lowest concentration is typically reached between 2-3 pm with an average concentration below 10 µg/m3. This daily pattern also adds uncertainties as well as possibilities to the effect of near-road greenspace on reducing $PM_{2.5}$ pollution.

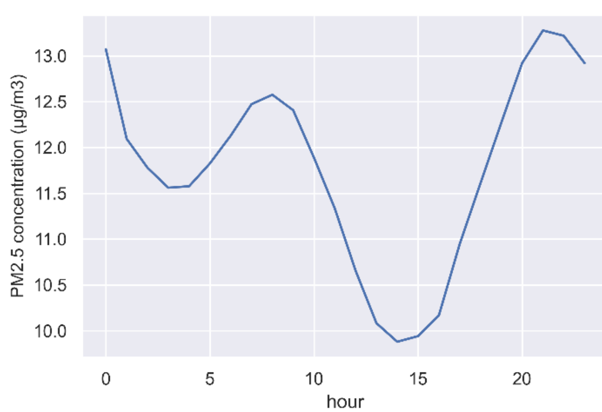


Figure 3: Diurnal change in the $PM_{2.5}$ concentration in London

The summary statistics for the dependent variable are shown down in table 2 together with the explanatory variables.

Spatial geometry manipulation

The 1km-radius buffers around all sites were generated based on site locations using Python package geopandas. Then all greenspaces and different types of roads in each site buffer were found using package shapely. The classification of roads contains the five types (including motorways) as mentioned above, as well as a sixth class 'Other' which represents all roads that are not assigned a road classification at national or local level (labelled 'Not Classified') or do not have the classification information (labelled 'Unknown') (OS, 2017). **Figure 4** shows an example of the site buffer as well as the roads and green spaces that are in it.

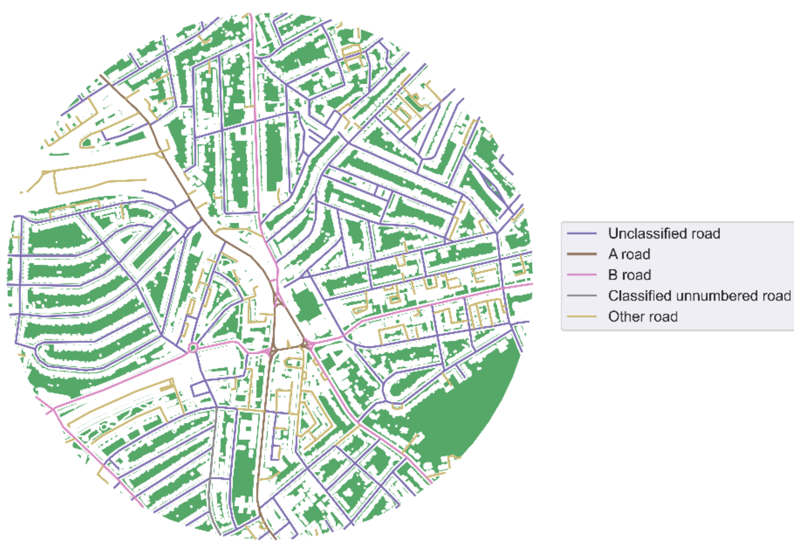


Figure 4: site buffer of CD1 (Camden - Swiss Cottage) as an example.

It is worth mentioning that there were two pairs of sites that were very close to each other (less than 1.5km). However, after performing a Student t-test (Kim, 2015) on their $PM_{2.5}$ data, it was found that they were significantly different (p-value < 0.05), hence they were kept in the study.

A 50m buffer zone was then generated for each road and used to find all near-road greenspaces. Several studies (Kassomenos *et al.*, 2014; Eeftens *et al.*, 2012; Holguin *et al.*, 2007) on road traffic and $PM_{2.5}$ pollution backed up the choice of a 50m buffer. These near-road greenspaces were also classified according to their nearby roads. For an area of green space that was close to more than one type of road, it was counted multiple times as near-road greenspace. This means that for each road class there is a set of marked near-road green spaces, and an area of green space can be marked as several different types of near-road greenspaces at the same time.

Generating explanatory variables

In order to investigate the effects of different types of near-road greenspace on reducing $PM_{2.5}$ concentration, it is important to determine the proper variables to represent the near-road greenspace conditions in each site buffer. The simplest possible choice would be to use the percentage of near-road greenspace, which is the area of near-road greenspaces divided by the area of all green spaces in a site buffer. The problem with using the area percentage as an indication is that for places where there are only a small number of roads the percentage will be very small while for places with many roads the percentage will be very large. As a result, instead of being an indication of near-road greenspace condition, it is actually an indication of the number of roads.

One approach to mitigate the influence of the number of roads is to divide the area of near-road greenspaces by the total length of roads. In this way, the division result becomes near-road greenspace area per road length, which only reflects the conditions of near-road sections of the greenspace regardless of the number of roads in each site buffer.

Therefore, in this study, a greenspace-area-per-road-length number was calculated for each type of road as an indicator for the near-road greenspace conditions in each site buffer. This was accomplished based on the road and greenspace geometries prepared in the previous section. All site buffers with no specific type of roads were assigned zero for their indicator for the specific road type.

Variable Name	Description	mean	std	min	25%	50%	75%	max
UnC_area_per_len	Unclassified road	28.52	14.02	6.46	13.89	32.96	38.75	47.13
A_area_per_len	A road	26.25	16.16	7.08	12.6	21.63	35.94	60.89
B_area_per_len	B road	25.79	21.17	0	5.63	27.49	44.13	56.16
CUn_area_per_len	Classified unnumbered road	34.01	22.54	0	17.72	33.96	50.73	64.51
Other_area_per_len	Other road	42.5	21.03	16.42	26.38	37.12	55.77	84.76
Value	$PM_{2.5}$ concentration	11.78	10.66	0.1	5.5	8.6	14	592.8

Table 2: Summary statistics of the explanatory variables and the PM2.5 data.

The summary statistics for the explanatory variables are shown in **table 2**. It is obvious that there are large variations within the same indicator of near-road greenspaces, which suggests a great difference between the conditions of near-road greenspaces in different areas. Therefore it becomes more practically relevant to study the relationship between the near-road greenspace conditions and regional $PM_{2.5}$ concentrations.

2.4 Multivariate linear regression models

A preliminary analysis of annual mean concentrations for all 21 sites was first performed to evaluate their overall relationship with the near-road greenspace. The global Moran's I of the annual means indicated that there was no obvious spatial auto-correlation in the dependent variable, so a non-spatial multivariate linear regression was performed.

The model performance was evaluated through a LOOCV (LeaveOneOut cross-validation) which is a type of cross-validation method that works well with small sample size data (Scikit-learn, 2013a). A typical cross-validation (k-fold) splits a dataset into k subsets and uses each subset once as the testing set to evaluate the performance of the model trained by all the other subsets (Scikit-learn, 2013b). The result of cross-validation is the average performance of the k models. When the sample size is small, the k-fold cross-validation result can have a large variance because how the samples are split will greatly alter the result. A LOOCV, on the other hand, split the samples into training and testing sets N times, where N is the sample size, with only one sample as the testing set and all the other N-1 samples as the training set. Each time the testing sample is used to evaluate the performance of the model fitted with the training set, and the cross-validation result is the average performance of the N models. The advantage of LOOCV is that the estimation is deterministic, meaning that there is no variance in the estimated performance of the model because every sample is used once to evaluate the model, and the process can be repeated (Wei *et al.*, 2019). The downside of LOOCV is its high computational cost (Syed, 2011; Wei *et al.*, 2019), although for a small sample size it is neglectable.

The effect of each type of near-road classification was determined according to their corresponding feature importance and model coefficient. The feature importance was computed using the permutation feature importance technique from the Python package sklearn, which calculates the decrease in the model performance when the specific feature (independent variable) is shuffled (Breiman, 2001). A common method is to repeat the shuffle procedure several times (in this case 50 times), and calculate the mean and the standard deviation from all the repeated samples for each feature. The feature importance reflects how much a model depends on a feature, and in the case of this study, how much effect each type of near-road greenspace has on the regional $PM_{2.5}$ level.

After the initial investigation of the relationship between near-road greenspace and regional $PM_{2.5}$ level, the temporal changes in it were explored in further depth. This was accomplished using a series of multivariate linear regression models. The temporal changes were analysed along with two time series: 12 months throughout a year and 24 hours throughout a day. The hourly $PM_{2.5}$ concentrations were first used to generate monthly mean concentrations as well as average concentrations at each hour during the year. The two sets of concentrations at different time intervals were then analysed in groups separated by each unit time interval. In other words, 12 groups of monthly mean concentrations and 24 groups of hourly mean concentrations on an average day during the year were analysed independently. Each analysis included an identification of spatial auto-correlation, a fit to a multivariate linear regression model, a LOOCV for model performance, permutation feature importance and a check for residual's normality. None of the variables was transformed or scaled, because the coefficients of the model were to be used as an indication of the effect of the near-road greenspace on $PM_{2.5}$ concentration.

3. Results

3.1 Modelling annual mean $PM_{2.5}$ concentration

The preliminary analysis of the annual mean $PM_{2.5}$ concentrations found a global Moran's I of 0.096 using a Gaussian kernel weights matrix, and the multivariate linear regression model as a function of the near-road greenspace conditions had an r-squared value of 0.365 and a LOOCV r-squared of 0.081. The residuals of the model were normally distributed and not spatially auto-correlated. **Table 3** shows the coefficient and feature importance mean and error estimations for the explanatory variables. The indicator for greenspace near Unclassified road had the highest estimated feature importance, which was even higher than the r-squared value of the model. This means that the performance depends heavily on the variable, and shuffling it would alter the r-squared value to negative. The effect of near-Unclassified-road greenspace, therefore, was the strongest among the five types, and the estimated coefficient (-0.108) indicated that the higher the indicator (near-Unclassified-road greenspace area per road length), the lower the $PM_{2.5}$ level. The B road and Classified unnumbered road indicators had median levels of estimated importance among the five, with similar coefficient estimations (0.038 for the former and 0.039 for the latter), which represented the positive effects of these two types of greenspaces on $PM_{2.5}$ concentrations. The feature importance estimations of the rest two indicators (A road and Other road) were both comparatively low, but their estimated effects were opposite. A road indicator had a positive effect (0.030) while Other road indicator had a negative effect (-0.025).

Road type	Coefficient	Feature importance	Std of feature importance
Unclassified road	-0.108	1.636	0.552
A road	0.030	0.195	0.107
B road	0.038	0.490	0.228
Classified unnumbered road	0.039	0.614	0.244
Other road	-0.025	0.200	0.135
R-squared		0.365	
LOOCV R-squared		0.081	

Table 3: Multivariate linear regression model for the annual mean $PM_{2.5}$ concentration as a function of the indicators for greenspaces near five types of roads. The sample size of the model is 21.

However, the low LOOCV r-squared value of the model and the high variations (high standard deviation) in the feature importance estimations made the result less convincing. There might be some spatial-temporal changes in the relationship that were altering the annual mean model performance, and since the potential influence from spatial auto-correlation had already been excluded, it was necessary to check the temporal changes.

3.2 Modelling annual mean $PM_{2.5}$ concentrations for each hour

The 24 groups of annual mean $PM_{2.5}$ concentrations for each hour were separately tested for global Moran's I and none of them was found spatial auto-correlated. They were then used to fit a multivariate linear regression model as a function of their near-road greenspace indicators, and their residuals were normally distributed with no spatial auto-correlation spotted. Their performances are shown in **Figure 5**. The models of the annual means for hours between 0 am and 3 am and between 9 am and 12 pm had better performance than the others, with an r-squared value higher than 0.4 and LOOCV r-squared value higher than 0.1. These hourly intervals were exactly the intervals at which the annual mean $PM_{2.5}$ concentrations were falling (Figure 3). The model with the highest performance was for 9 am where the r-squared (0.481) and the LOOCV r-squared value (0.227) were both the highest out of the 24 models. Conversely, the hourly intervals when the corresponding model performance was low coincided with the time periods when the concentrations were rising. The lowest-performance model was for 5 pm with an r-squared value of 0.203 and a LOOCV r-squared value of 0.005. The average r-squared value of the 24 models was 0.337 and the average LOOCV r-squared value was 0.065.

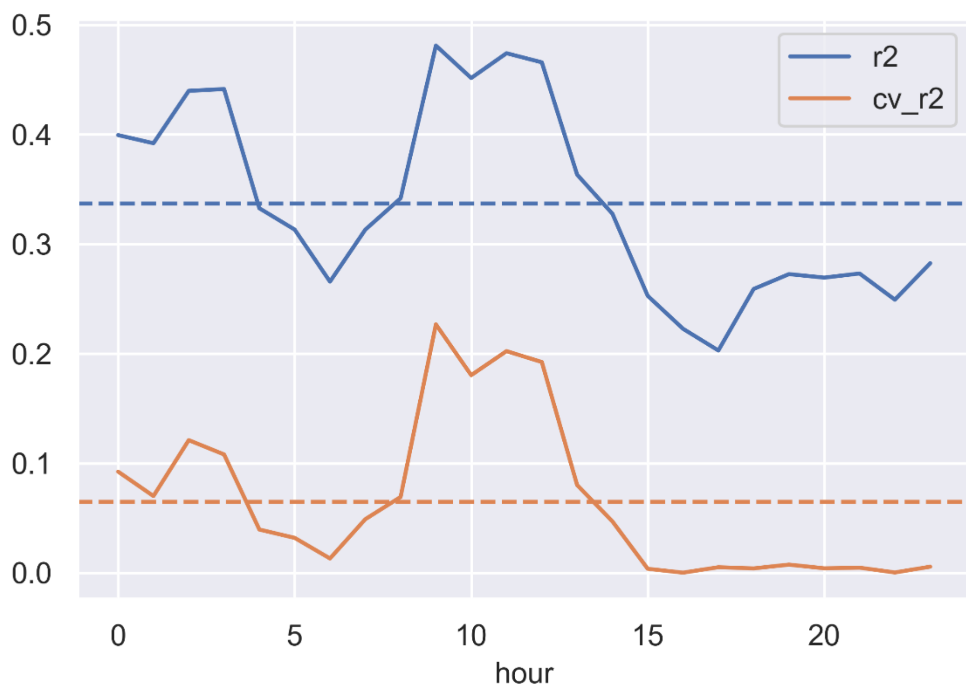


Figure 5: Performance of the 24 models in terms of r-squared value and LOO cross-validation r-squared value. The two dashed lines indicate the average r-squared and LOOCV r-squared values of the 24 models.

The feature importance estimations for models of all hourly intervals are presented in **Figure 6**. In almost all models the indicator for greenspaces near Unclassified roads had the highest feature importance, especially in the high-performance models where its importance was significantly higher than the other four features. In contrast, A road and Other road greenspace indicators had the two lowest estimated feature importance (between 0.2 and 0.3) in most models except for 0 am-4 am when the estimation for A road was relatively higher and 5 am-8 am when that for Other road was around 0.5. The indicators for B road and Classified unnumbered road had comparatively stable feature importance with a value between 0.3 and 0.6 in most models. The variations in the estimations were still relatively large, with a higher average deviation (0.41) for the Unclassified road indicator and lower for the rest.

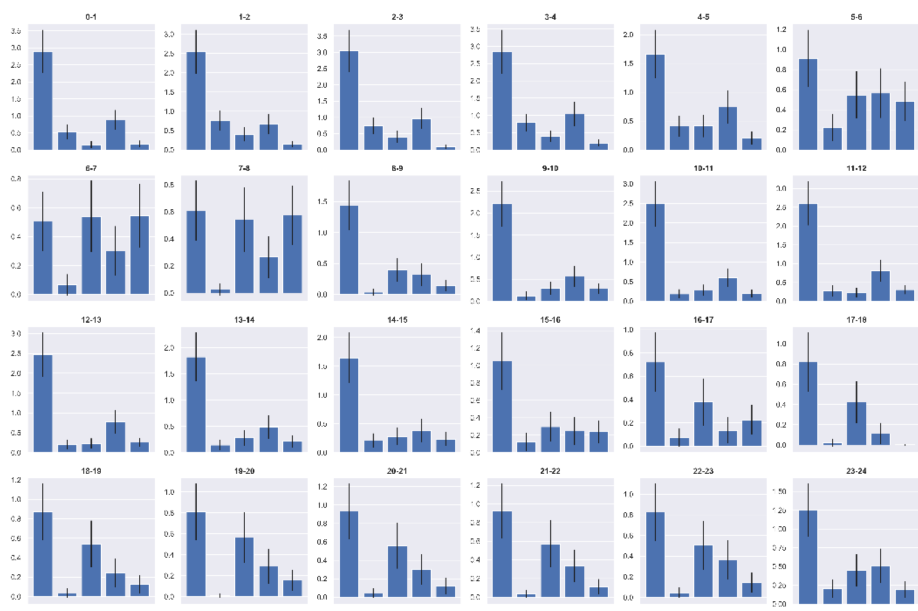


Figure 6: Feature importance estimations for the 24 models. The five bars from left to right represent: Unclassified road, A road, B road, Classified unnumbered road, Other road. The error bar is presented in black line, which is generated from the calculated standard deviation of each feature importance estimation.

The model coefficients were used to determine the effect of different indicators, and hence different types of near-road greenspace, on $PM_{2.5}$ levels. **Figure 7** shows the coefficients of all indicators for every model. In all 24 models, the effects of greenspaces near Unclassified and Other roads on $PM_{2.5}$ levels were negative while the other three types of greenspaces had positive effects. The changing patterns in the effect sizes of the five features as well as the intercept were relatively similar, which was a declining-rising-declining-rising-declining trend.

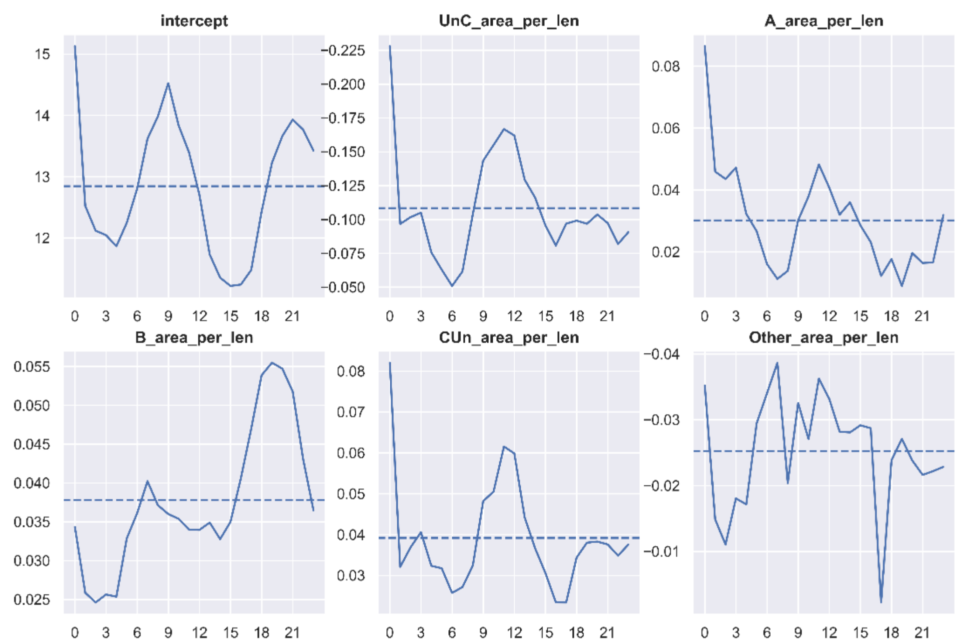


Figure 7: Intercepts and coefficients of all features for the 24 models. The dashed lines represent the average of the coefficients from the 24 models.

3.3 Modelling monthly mean $PM_{2.5}$ concentrations

The same routine of analyses was performed on the 12 groups of monthly mean $PM_{2.5}$ concentrations. Since no spatial auto-correlation was spotted in all groups of monthly means, multivariate linear regression models as a function of near-road greenspace conditions were fitted and tested. The residuals for all 12 models were approximately normally distributed, and no spatial auto-correlation was found in the residuals. The model evaluation results are shown in **Figure 8**. The models for the monthly means during the summer months (June to August) overall performed better than the others. The June model had the highest r-squared value (0.545) as well as the highest LOOCV r-squared value (0.280). The second-highest performing model was the August one with an r-squared value of 0.499 and a LOOCV r-squared value of 0.189. On the other hand, winter and spring month models had a lower performance. The model for April had the lowest r-squared value of 0.079 with a LOOCV r-squared value of 0.102. The high- and low-performance seasons coincided with the high and low seasons of $PM_{2.5}$ as shown in Figure 2. The r-squared mean for the 12 months was 0.323 and the LOOCV r-squared mean was 0.079.

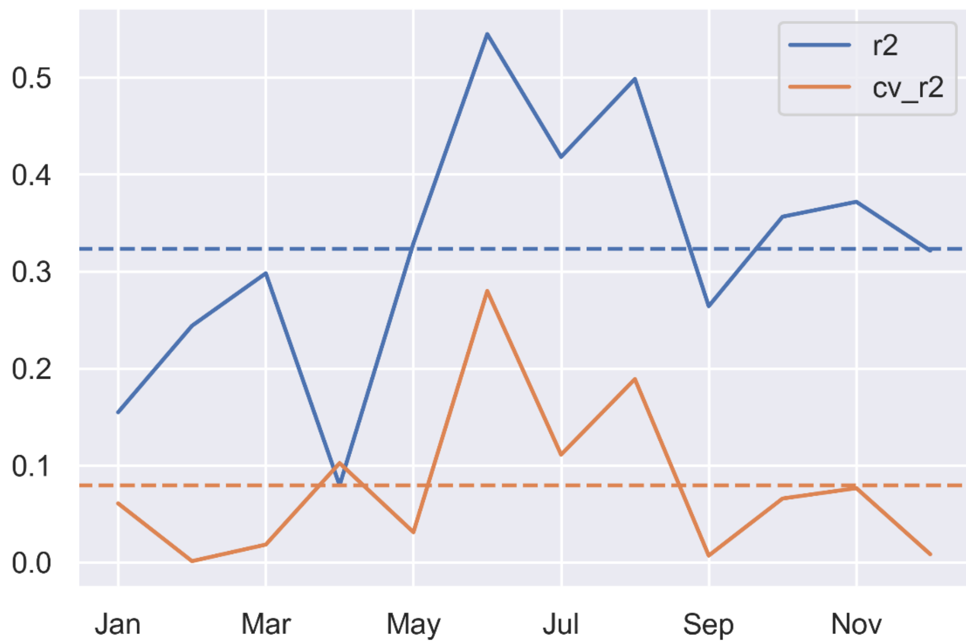


Figure 8: Performance of the 12 models. The dashed lines indicate the average performance of the 12 models.

As for the feature importance estimation, The Unclassified road indicator had the highest average importance, as shown in **Figure 9**. It was significantly higher than other features from February to July, but was lower during September to November, and almost to the same level as B road in August and during December to January. A road had low feature importance estimations in most months, except for June when it was estimated to be the second most important feature with an estimation of 1.932. Similarly, Other road had relatively low estimations from January to July and November, with high estimations from August to October and December. B road and Classified unnumbered road both had low feature importance estimations between February and June, while estimations for July, September and December were median. In the models for August, October and November, estimations for B road were relatively high while estimations for Classified unnumbered road were low.

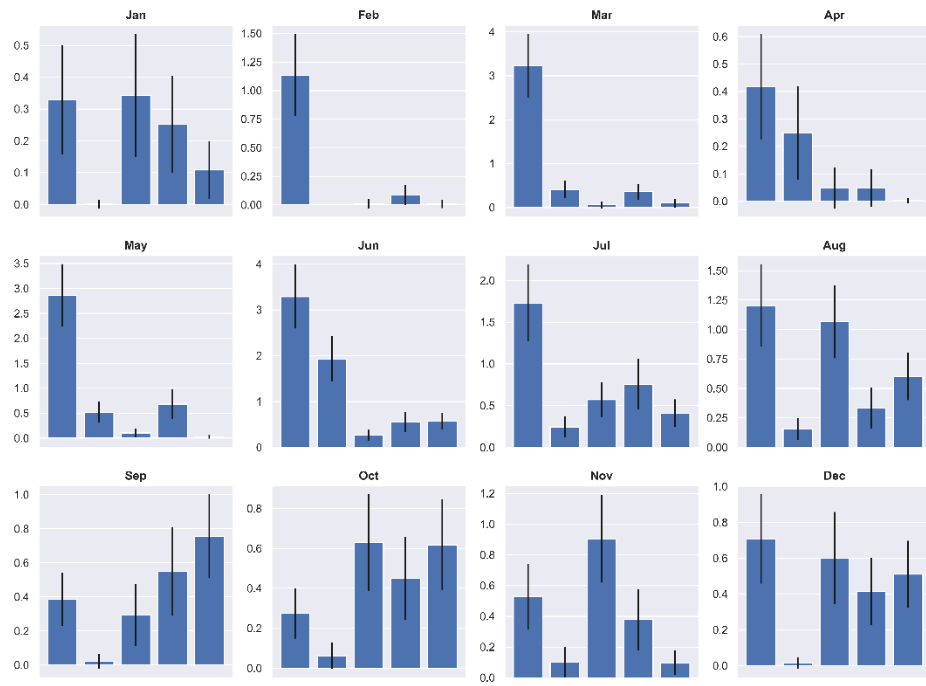


Figure 9: Feature importance estimations for the 12 models. The five bars from left to right represent: Unclassified road, A road, B road, Classified unnumbered road, Other road. The error bar is presented in black line, which is generated from the calculated standard deviation of each feature importance estimation.

The variations in the estimated feature importance were overall lower than that in the 24 groups of annual means for each hour. The average standard deviation of the estimations for Unclassified road indicator was the highest (0.361), with the other four indicators having about half the average standard deviation.

The patterns of the coefficients are presented in **Figure 10**. The changes in the intercepts of the 12 models followed a similar pattern to the monthly $PM_{2.5}$ concentrations as shown in Figure 2, with peaked values in February and April. The coefficients of the five features were much more chaotic. On one hand, the effect of Unclassified road indicator remained negative for the entire year. Its size was higher from February to August and peaked (-0.205) in June, while lower during autumn (September to November) and early winter (December and January). On the other hand, the effects of B road and Classified unnumbered road indicators remained positive, with the effect size of the former peaking in August and the latter from May to July. The rest two features, being the indicators for A road and Other road, had relatively higher variations in their model coefficients. The effect of A road was positive in most months, with the size of the effect reaching its maximum in June at 0.137. However, its effect in October and November was negative, although the effect size was not large. Conversely, the effect of Other road was negative except for in February and March, with the highest negative effect size in June at -0.054 and the highest positive effect size in March at 0.020. Overall, the average effect size of the Unclassified road was the highest (-0.109) and that of the Other road was the lowest (-0.025).

Figure 10: Intercepts and coefficients of all features for the 12 models. The dashed lines represent the average of the coefficients from the 12 models.

4. Discussion

4.1 The indicators of the greenspaces near different types of roads

The explanatory variables of the models used in this study were the total greenspace area near each type of road divided by the total length of that type of road (for each site buffer). This is an indication of the condition of the greenspaces for the specific type of road, and a higher value means a higher coverage of near-road greenspaces. However, such an indicator does not represent any information about the area of near-road greenspaces, hence a higher value of the indicator does not mean more near-road greenspaces for a specific type of road in a site buffer. Nor does it represent a relationship between the near-road greenspaces between two roads; for example, a higher indicator of near-road greenspace for one road type than that for the other type within the same buffer can only represent a higher greenspace coverage around the former road than the latter within that buffer, and not any information beyond that.

As a result, the detected relationships between those explanatory variables and regional $PM_{2.5}$ concentrations represent how the greenspace coverage near a specific type of road affects the local $PM_{2.5}$ levels (i.e. higher coverage for a specific road type will result in a higher or lower $PM_{2.5}$ level). Since the levels of coverage of the different near-road greenspaces can have different effects on the local $PM_{2.5}$ concentrations, this can also be referred to as the effect of the near-road greenspace for a specific road type on regional $PM_{2.5}$ levels.

4.2 Overall effects of near-road greenspaces in changing regional $PM_{2.5}$ levels

The results from the model for the annual mean $PM_{2.5}$ concentration highlight that there is a strong negative relationship between the indicator for greenspaces near Unclassified roads and the regional $PM_{2.5}$ levels, with the size of the effect being the highest among the five types of roads. Its feature importance is also estimated to be the highest, indicating the highest impact on the $PM_{2.5}$ levels. On the other hand, the results suggest a positive effect of the coverage of greenspaces near B roads and Classified unnumbered roads on regional $PM_{2.5}$ levels, which means higher coverage of near-road greenspaces is related to a higher level of $PM_{2.5}$. This indicates that instead of reducing $PM_{2.5}$ pollution, these green spaces are actually intensifying it. The moderate feature importance estimations for the two indicators imply that the near-road greenspaces for the two types of road have a median impact on regional $PM_{2.5}$ levels. As for A road, the predicted effect is also positive, and the estimated feature importance suggests that the impact of greenspaces near A roads is even lower than that of B road and Classified unnumbered road. The coverage of greenspaces near Other roads is estimated to have a weak negative effect on regional $PM_{2.5}$ levels with low estimated feature importance.

The higher impact (feature importance) of the greenspaces near Unclassified roads is likely due to the difference in the average traffic volume of different types of roads. The Unclassified roads generally have a lower traffic volume than A roads, B roads and Classified unnumbered roads (Roads.org.uk, 2017; GOV.UK, 2012). Therefore, the greenspaces near Unclassified roads are normally exposed to lower $PM_{2.5}$ particle intensities than those near the other types of roads. This typically results in a better function of the leaf in storing $PM_{2.5}$ particles over a longer period of time. When air containing particulate matter passes through a green space, part of the particulate matter will be removed from the air by the leaf surface and the wax layer on it (Kończak *et al.*, 2021). If the air contains a large amount of particulate matter, the leaf will soon reach its maximum in storing them (Liu *et al.*, 2013) and maintains a low PM removal efficiency before its recovery through the wind (Schaubroeck *et al.*, 2014) and rainfall wash-off (Weerakkody *et al.*, 2018; Xu *et al.*, 2020; Schaubroeck *et al.*, 2014; Xu *et al.*, 2017). Conversely, if the air only contains a small amount of PM, the leaf can capture particles and remove them from its surface at the same time. In this case, the green space will have an overall higher efficiency in capturing PM. The coverage of greenspaces near Unclassified roads that are exposed to a lower amount of $PM_{2.5}$ hence have a higher effect on the regional $PM_{2.5}$ levels. Similarly, B road and Classified unnumbered road have lower traffic volumes than A road so they have higher impacts on regional $PM_{2.5}$ levels.

In terms of the division between the positive and negative effects of coverages of greenspaces near different types of roads on the $PM_{2.5}$ concentrations, the determinants are more complicated. On one hand, greenspaces reduce regional $PM_{2.5}$ pollution by removing particles from the air (Beckett *et al.*, 2000; Nowak *et al.*, 2006; Kończak *et al.*, 2021) as well as blocking its transmission (Hofman *et al.*, 2016; Morakinyo and Lam, 2016). On the other hand, the block of transmission also results in a higher regional concentration (Morakinyo and Lam, 2016). As Vos *et al.* (2013) suggested, at least locally, greenspaces near roads have negative effects on reducing pollutants, because the existence of the vegetation reduces the ventilation, and therefore the pollutants accumulate in the area. This aerodynamic especially adds to pollution in street canyons where there are built-ups on both sides of the road. The trees slow the wind speed in a street canyon and reduce the exchange between the air within the canyon and above the roof, which results in an accumulation of pollution (Abhujith *et al.*, 2017; Jeanjean *et al.*, 2017). As a result, the effect of green space on regional $PM_{2.5}$ levels depends on the combination of the removal and the increase of the particulate matter. The results from the model suggest that the overall removal effect of PM exceeds the intensification effect for greenspaces near Unclassified roads and Other roads, and the intensification effect surpasses the removal effect for the other three types of roads. The specific reasons behind such a pattern need some further investigation. One proposed conjecture is that this is due to morphological differences. A roads, B roads and Classified unnumbered roads are those higher-class roads that form the main network, and are therefore connected to more nearby built-ups. Hence the street canyon pattern is more common on those roads than on Unclassified roads and Other roads, so the near-road greenspaces on those roads tend to increase the regional $PM_{2.5}$ concentration.

4.3 Temporal changes in the effect of near-road greenspaces

The results from the two groups of models that examined the temporal changes in the relationships between the coverage of near-road greenspaces for different types of roads and regional $PM_{2.5}$ levels highlight the significant role of greenspaces near Unclassified roads in reducing $PM_{2.5}$ pollution. In most scenarios greenspaces near Unclassified roads have the highest influence on regional $PM_{2.5}$ levels. This is especially true from February to July on a monthly basis, and before 5 am and after 8 am on a daily basis, when its estimated impact is significantly higher than the greenspaces near the other four types of roads.

The estimated effect of greenspaces near Unclassified roads is negative in all periods. In terms of monthly variations, the relatively high estimated effect from May to July aligns with results from other studies (Wang *et al.*, 2015; Jeanjean *et al.*, 2017; Xu *et al.*, 2017; Steffens *et al.*, 2012) that found an overall higher reducing effect of vegetation on PM levels during summer. This is partly due to the fact that the leaf area density is higher in summer (Jeanjean *et al.*, 2017; Steffens *et al*, 2012) so more PMs are deposited on the leaf surface. Other factors include a higher frequency of rainfall (Xu *et al.*, 2017; Wang *et al.*, 2015), higher wind speed and more vertical air exchange due to higher solar radiation (Wang *et al.*, 2015). In contrast, the lowest estimated effect is from September to January, which also coincides with a series of other studies (Przybysz *et al.*, 2018; Przybysz *et al.*, 2014; He *et al.*, 2020) that observed a lower reducing effect of vegetation on PM pollution during winter mainly due to lower wind speed, cooler temperature and less precipitation.

As for the diurnal variations, the highest effect of greenspaces near Unclassified roads on $PM_{2.5}$ concentration was observed at midnight and a relatively high effect during 9-12 am, and the lowest effect was during 5-7 am. There are very few studies on the diurnal variations in the PM removal effect of vegetation (Deng *et al.*, 2019; Brantley *et al.*, 2014), and neither of them found a significant change in the effect throughout an average day. The possible cause of such detected variation may be the change in temperature and solar radiation, and the resulting wind speed change. When the sun rises, the increasing temperature creates more convections, which generate stronger wind that washes off the leaf surface. However, such speculation can not explain the peak in the effect at midnight, and therefore the diurnal variation may need further investigations.

The effect of near-road greenspaces for Other roads is similar to that for Unclassified roads, except for in February and March when it is positive, and overall much lower influence on $PM_{2.5}$ levels. Nevertheless, the effect of greenspaces near A roads, B roads and Classified unnumbered roads is, as estimated for the overall effect, positive in virtually all periods. This suggests that the intensification effect of these green spaces on PM pollution surpasses the removal effect all the time. The positive effect of all three peaks in summer and troughs in winter or spring, with the effect of A road greenspaces, shortly being negative during October to November.

The peaked positive effect in summer is likely due to the higher temperature and the resulting low pressure around the road that prevents airborne PM from leaving by creating inward airflow (Al-Dabbous and Kumar, 2014). The presence of near-road greenspaces strengthens such an effect by lowering air circulation at the surface level whose consequence is much more impactful than the removing capacity of the vegetation leaves, especially in street canyons (Jeanjean *et al.*, 2017). Similarly, the low positive (and even negative) effect in winter and spring can be explained by the high-pressure climate that promotes the outward flow of near-surface air in street canyons, and therefore reduce the intensification effect of green spaces on regional $PM_{2.5}$ levels (Al-Dabbous and Kumar, 2014). This theory can also be implied to the diurnal variations where the high effect occurs during 9 am-2 pm and the low effect during nighttime. The exceptions are during 6-9 pm for B road and at midnight for A road and Classified unnumbered road when the effect is very high. The former can be explained by higher traffic flows during the period, but the cause of the latter still needs some further investigation.

5. Conclusion

This paper examined the effect of near-road greenspaces for different types of roads on regional $PM_{2.5}$ levels and the temporal changes in the effect through modelling the relationships between coverages of near-road greenspaces and local $PM_{2.5}$ concentrations. This effect is determined by the combination of removal ability and intensification ability. Several conclusions were drawn from this study:

1. The greenspaces near Unclassified roads have the highest overall reducing effect on regional $PM_{2.5}$ levels, with a relatively high effect in summer and from 9-12 am on an average day. The potential pushes are most possibly the higher leaf area density and warmer temperatures.

2. The A road, B road and Classified unnumbered road greenspaces have an opposing effect to reduce $PM_{2.5}$ levels. The most likely reason is that street canyon morphology, which tends to create intensified PM pollution, is much more common around these roads. The positive effect generally peaks seasonally in summer and diurnally during 9-12 am.

3. The temporal change in the effect of Other road greenspaces is most varied. The overall effect on regional $PM_{2.5}$ levels is negative, but in spring the effect is positive.

4. Except for B road, greenspaces near all the other four types of roads have a high effect (either positive or negative) on $PM_{2.5}$ level at midnight.

The aforementioned points derived from this study should lead to some practical policy implications in urban greenspace planning. It proves that greenspace, especially near-road greenspace, is not guaranteed to reduce PM pollution in all circumstances. However, due to some constraints of this study, including the small sample size and failure in explaining some of the temporal changes, some further explorations should be made for a better understanding of the function of green spaces, and for more cost-effective implementations of urban greenspaces.

Reference List

- Abhijith, K. V., Kumar, P., Gallagher, J., McNabola, A., Baldauf, R., Pilla, F., Broderick, B., Sabatino, S. D. and Pulvirenti, B. (2017) Air pollution abatement performances of green infrastructure in open road and built-up street canyon environments–A review. *Atmospheric Environment*, *162*, 71-86.
- Al-Dabbous, A. N. and Kumar, P. (2014) The influence of roadside vegetation barriers on airborne nanoparticles and pedestrians exposure under varying wind conditions. *Atmospheric Environment*, *90*, 113-124.
- Anderson, J. O., Thundiyil, J. G. and Stolbach, A. (2012) Clearing the air: a review of the effects of particulate matter air pollution on human health. *Journal of medical toxicology*, *8*(2), 166-175.
- Baldauf R., Jackson L., Hagler G., Isakov V., McPherson G., Cahill T.A., Zhang K.M., Cook J.R., Bailey C. and Wood P. (2011) The role of vegetation in mitigating air quality impacts from traffic emissions. *EM. Jan: 1-3*.
- Beckett, K. P., Freer-Smith, P. H. and Taylor, G. (2000) Particulate pollution capture by urban trees: effect of species and windspeed. *Global Change Biology*, *6*(8), 995–1003.
- Brantley, H. L., Hagler, G. S. W., J. Deshmukh, P. and Baldauf, R. W. (2014) Field assessment of the effects of roadside vegetation on near-road black carbon and particulate matter. *Science of the Total Environment*, *468-469*, 120–129.
- Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.
- Brook, R.D., Rajagopalan, S., Pope III, C.A., Brook, J.R., Bhatnagar, A., Diez-Roux, A.V., Holguin, F., Hong, Y., Luepker, R.V., Mittleman, M.A. and Peters, A. (2010) Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the American Heart Association. *Circulation*, *121*(21), 2331-2378.

- Cai, L., Zhuang, M. and Ren, Y. (2020) Spatiotemporal characteristics of NO2, PM2. 5 and O3 in a coastal region of southeastern China and their removal by green spaces. *International Journal of Environmental Health Research*, 1-17.
- Chen, X. and Xiao, Y. (2018) A Novel Method for Air Quality Data Imputation by Nuclear Norm Minimization. *Journal of Sensors*, *2018*, 1–11.
- Chen, M., Dai, F., Yang, B. and Zhu, S. (2019) Effects of neighborhood green space on PM2.5 mitigation: Evidence from five megacities in China. *Building and Environment*, *156*, 33–45.
- Defra. (2022) Automatic Urban and Rural Network (AURN). [Online]. Available from: https://uk-air.defra.gov.uk/networks/network-info?view=aurn [Accessed 06 Feb 2022]
- Deng, S., Ma, J., Zhang, L., Jia, Z. and Ma, L. (2019) Microclimate simulation and model optimization of the effect of roadway green space on atmospheric particulate matter. *Environmental Pollution*, *246*, 932–944.
- Digimap. (2021) Ordnance Survey. [Online]. Available from: https://digimap.edina.ac.uk/os [Accessed 18 Dec 2021]
- Eeftens, M., Beelen, R., De Hoogh, K., Bellander, T., Cesaroni, G., Cirach, M., Declercq, C., Dedele, A., Dons, E., De Nazelle, A. and Dimakopoulou, K. (2012) Development of land use regression models for PM2. 5, PM2. 5 absorbance, PM10 and PMcoarse in 20 European study areas; results of the ESCAPE project. *Environmental science & technology*, *46*(20), 11195-11205.
- González-Flecha, B. (2004) Oxidant mechanisms in response to ambient air particles. *Molecular aspects of medicine*, *25*(1-2), 169-182.
- GOV.UK. (2012) Guidance on road classification and the primary route network. [Online]. Available from: https://www.gov.uk/government/publications/guidance-on-road-classification-and-the-primary-route-network/guidance-on-road-classification-and-the-primary-route-network [Accessed 08 Feb 2022]

- Greenspace Information for Greater London (GiGL). (2018) Mapping London's Green Belt and Metropolitan Open Land. [Online]. Available from: https://www.gigl.org.uk/mapping-londons-green-belt-and-mol/ [Accessed 14 Feb 2022]
- He, C., Qiu, K., Alahmad, A. and Pott, R. (2020) Particulate matter capturing capacity of roadside evergreen vegetation during the winter season. *Urban Forestry & Urban Greening*, *48*, 126510.
- Hofman, J., Bartholomeus, H., Janssen, S., Calders, K., Wuyts, K., Van Wittenberghe, S. and Samson, R. (2016) Influence of tree crown characteristics on the local PM10 distribution inside an urban street canyon in Antwerp (Belgium): A model and experimental approach. *Urban forestry & urban greening*, *20*, 265-276.
- Holguin, F., Flores, S., Ross, Z., Cortez, M., Molina, M., Molina, L., Rincon, C., Jerrett, M., Berhane, K., Granados, A. and Romieu, I. (2007) Traffic-related Exposures, Airway Function, Inflammation, and Respiratory Symptoms in Children. *American Journal of Respiratory and Critical Care Medicine*, *176*(12), 1236–1242.
- Irga, P. J., Burchett, M. D. and Torpy, F. R. (2015) Does urban forestry have a quantitative effect on ambient air quality in an urban environment?. *Atmospheric Environment*, *120*, 173-181.
- Jeanjean, A. P. R., Buccolieri, R., Eddy, J., Monks, P. S. and Leigh, R. J. (2017) Air quality affected by trees in real street canyons: The case of Marylebone neighbourhood in central London. *Urban Forestry & Urban Greening*, *22*, 41–53.
- Ji, W. and Zhao, B. (2014) Numerical study of the effects of trees on outdoor particle concentration distributions. In *Building Simulation*. 7(4), 417-427. Tsinghua University Press.
- Karagulian, F., Belis, C. A., Dora, C. F. C., Prüss-Ustün, A. M., Bonjour, S., Adair-Rohani, H. and Amann, M. (2015) Contributions to cities' ambient particulate matter (PM): A systematic review of local source contributions at global level. *Atmospheric environment*, *120*, 475-483.

- Kassomenos, P. A., Vardoulakis, S., Chaloulakou, A., Paschalidou, A. K., Grivas, G., Borge, R. and Lumbreras, J. (2014) Study of PM10 and PM2.5 levels in three European cities: Analysis of intra and inter urban variations. *Atmospheric Environment*, *87*, 153–163.
- Kim, T.K. (2015) T test as a parametric statistic. *Korean Journal of Anesthesiology, 68*, 540 - 546.
- Kim, H. (2020) Land use impacts on particulate matter levels in Seoul, South Korea: comparing high and low seasons. *Land*, *9*(5), 142.
- Kim, K.-H., Kabir, E. and Kabir, S. (2015) A review on the human health impact of airborne particulate matter. *Environment International*, *74*, 136–143.
- Kończak, B., Cempa, M. and Deska, M. (2021) Assessment of the ability of roadside vegetation to remove particulate matter from the urban air. *Environmental Pollution*, *268*, 115465.
- Lei, Y., Duan, Y., He, D., Zhang, X., Chen, L., Li, Y., Gao, Y.G., Tian, G. and Zheng, J. (2018) Effects of urban greenspace patterns on particulate matter pollution in metropolitan Zhengzhou in Henan, China. *Atmosphere*, *9*(5), 199.
- Li, R., Li, Z., Gao, W., Ding, W., Xu, Q. and Song, X. (2015) Diurnal, seasonal, and spatial variation of PM 2.5 in Beijing. *Science Bulletin*, *60*(3), 387–395.
- Liu, Z., Hu, B., Wang, L., Wu, F., Gao, W. and Wang, Y. (2014) Seasonal and diurnal variation in particulate matter (PM10 and PM2.5) at an urban site of Beijing: analyses from a 9-year study. *Environmental Science and Pollution Research*, *22*(1), 627–642.
- Liu, X., Yu, X. and Zhang, Z. (2015) PM2. 5 concentration differences between various forest types and its correlation with forest structure. *Atmosphere*, *6*(11), 1801-1815.
- Liu, L., Guan, D., Peart, M. R., Wang, G., Zhang, H. and Li, Z. (2013) The dust retention capacities of urban vegetation—a case study of Guangzhou, South China. *Environmental Science and Pollution Research*, *20*(9), 6601-6610.

- Löndahl, J., Pagels, J., Swietlicki, E., Zhou, J., Ketzel, M., Massling, A. and Bohgard, M. (2006) A set-up for field studies of respiratory tract deposition of fine and ultrafine particles in humans. *Journal of Aerosol Science*, *37*(9), 1152-1163.
- London Air. (2022) About Londonair. [Online]. Available from: https://www.londonair.org.uk/LondonAir/General/about.aspx [Accessed 06 Feb 2022]
- London Air. (2022b) Data Downloads. [Data]. Available from: https://www.londonair.org.uk/london/asp/datadownload.asp [Accessed 03 Jan 2022]
- London Datastore. (2019) Air Quality Monitoring Sites. [Data]. Available from: https://data.london.gov.uk/download/air_quality_monitoring_sites/b22020b1-c42a-4653-afb0-7f52fd658635/Air_Quality_Monitoring_Sites.gpkg [Accessed 12 Dec 2021]
- Mayor of London. (2019) PM2.5 in London: Roadmap to meeting WHO guidelines by 2030. [Report]. Available from: https://www.london.gov.uk/sites/default/files/pm2.5_in_london_october19.pdf [Accessed 06 Feb 2022]
- Morakinyo, T. E. and Lam, Y. F. (2016) Simulation study of dispersion and removal of particulate matter from traffic by road-side vegetation barrier. *Environmental Science and Pollution Research*, *23*(7), 6709-6722.
- Norazian, M. N., Shukri, Y. A., Azam, R. N. and Al Bakri, A. M. M. (2008) Estimation of missing values in air pollution data using single imputation techniques. *ScienceAsia*, *34*(3), 341-345.
- Nowak, D. J., Crane, D. E. and Stevens, J. C. (2006) Air pollution removal by urban trees and shrubs in the United States. *Urban forestry & urban greening*, *4*(3-4), 115-123.
- Ordnance Survey (OS). (2017) OS Open Roads User guide and technical specification. [Report]. Available from:

https://www.ordnancesurvey.co.uk/documents/os-open-roads-user-guide.pdf [Accessed 02 Feb 2022]

- Ordnance Survey (OS). (2021a) OS MasterMap Greenspace - Layer. [Online]. Available from: https://www.ordnancesurvey.co.uk/business-government/products/mastermap-greenspace [Accessed 17 Dec 2021]
- Ordnance Survey (OS). (2021b) OS Open Roads. [Online]. Available from: https://www.ordnancesurvey.co.uk/business-government/products/open-map-roads [Accessed 17 Dec 2021]
- Przybysz, A., Sæbø, A., Hanslin, H. M. and Gawroński, S. W. (2014) Accumulation of particulate matter and trace elements on vegetation as affected by pollution level, rainfall and the passage of time. *Science of the Total Environment*, *481*, 360–369.
- Przybysz, A., Nersisyan, G. and Gawroński, S. W. (2018) Removal of particulate matter and trace elements from ambient air by urban greenery in the winter season. *Environmental Science and Pollution Research*, *26*(1), 473–482.
- Roads.org.uk. (2017). Road Numbers: Numbers for A and B-roads. [Online]. Available from: https://www.roads.org.uk/articles/road-numbers/numbers-and-b-roads [Accessed 12 Feb 2022]
- Ross, Z., Jerrett, M., Ito, K., Tempalski, B. and Thurston, G. (2007) A land use regression for predicting fine particulate matter concentrations in the New York City region. *Atmospheric Environment*, *41*(11), 2255–2269.
- Schaubroeck, T., Deckmyn, G., Neirynck, J., Staelens, J., Adriaenssens, S., Dewulf, J., Muys, B. and Verheyen, K., (2014) Multilayered modeling of particulate matter removal by a growing forest over time, from plant surface deposition to washoff via rainfall. *Environmental science & technology*, *48*(18), 10785-10794.
- Scikit-learn. (2013a) 3.1.2.1.3. Leave One Out (LOO). [Online]. Available from: https://scikit-learn.org/stable/modules/cross_validation.html#leave-one-out-loo [Accessed 09 Feb 2022]

- Scikit-learn. (2013b) 3.1.2.1.1. K-Fold. [Online]. Available from: https://scikit-learn.org/stable/modules/cross_validation.html#k-fold [Accessed 09 Feb 2022]
- Shah, A. S., Langrish, J. P., Nair, H., McAllister, D. A., Hunter, A. L., Donaldson, K., Newby, D. E. and Mills, N. L. (2013) Global association of air pollution and heart failure: a systematic review and meta-analysis. *The Lancet*, *382*(9897), 1039–1048.
- Song, Y., Maher, B. A., Li, F., Wang, X., Sun, X. and Zhang, H. (2015) Particulate matter deposited on leaf of five evergreen species in Beijing, China: Source identification and size distribution. *Atmospheric environment*, *105*, 53-60.
- Steffens, J. T., Wang, Y. J. and Zhang, K. M. (2012) Exploration of effects of a vegetation barrier on particle size distributions in a near-road environment. *Atmospheric Environment*, *50*, 120–128.
- Sun, Q., Hong, X. and Wold, L. E. (2010) Cardiovascular effects of ambient particulate air pollution exposure. *Circulation*, *121*(25), 2755-2765.
- Srimuruganandam, B. and Nagendra, S. S. (2012) Source characterization of PM10 and PM2. 5 mass using a chemical mass balance model at urban roadside. *Science of the total environment*, *433*, 8-19.
- Syed, A. R. (2011) A review of cross validation and adaptive model selection.
- The Independent. (2014) 47 per cent of London is green space: Is it time for our capital to become a national park? [Online] Available from: https://www.independent.co.uk/climate-change/news/47-per-cent-of-london-is-green-space-is-it-time-for-our-capital-to-become-a-national-park-9756470.html [Accessed 26 Feb 2022]
- Vos, P. E., Maiheu, B., Vankerkom, J. and Janssen, S. (2013) Improving local air quality in cities: to tree or not to tree?. *Environmental pollution*, *183*, 113-122.
- Wang, Q. and Li, S. (2021) Nonlinear impact of COVID-19 on pollutions – Evidence from Wuhan, New York, Milan, Madrid, Bandra, London, Tokyo and Mexico City. *Sustainable Cities and Society*, *65*, 102629.

- Wang, H., Shi, H. and Wang, Y. (2015) Effects of Weather, Time, and Pollution Level on the Amount of Particulate Matter Deposited on Leaves of *Ligustrum lucidum*. *The Scientific World Journal*, *2015*, 1–8.
- Wei, J., Chu, X., Sun, X., Xu, K., Deng, H., Chen, J., Wei, Z. and Lei, M. (2019) Machine learning in materials science. *InfoMat*, *1*(3), 338–358.
- Weerakkody, U., Dover, J. W., Mitchell, P. and Reiling, K. (2018) The impact of rainfall in remobilising particulate matter accumulated on leaves of four evergreen species grown on a green screen and a living wall. *Urban Forestry & Urban Greening*, *35*, 21–31.
- World Bank. (2017) PM2.5 air pollution, mean annual exposure (micrograms per cubic meter). [Online]. Available from: https://data.worldbank.org/indicator/EN.ATM.PM25.MC.M3?end=2017&start=2017&view=map [Accessed 03 Feb 2022]
- World Health Organization (WHO). (2021) Ambient (outdoor) air pollution. [Online]. Available from: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health [Accessed 03 Feb 2022].
- Wu, C. D., Chen, Y. C., Pan, W. C., Zeng, Y. T., Chen, M. J., Guo, Y. L. and Lung, S. C. C. (2017) Land-use regression with long-term satellite-based greenness index and culture-specific sources to model PM2. 5 spatial-temporal variability. *Environmental Pollution*, *224,* 148-157.
- Xu, X., Zhang, Z., Bao, L., Mo, L., Yu, X., Fan, D. and Lun, X. (2017) Influence of rainfall duration and intensity on particulate matter removal from plant leaves. *Science of the Total Environment*, *609*, 11–16.
- Xu, M., Sbihi, H., Pan, X. and Brauer, M. (2019) Local variation of PM2.5 and NO2 concentrations within metropolitan Beijing. *Atmospheric Environment*, *200*, 254-263.
- Xu, X., Xia, J., Gao, Y. and Zheng, W. (2020) Additional focus on particulate matter wash-off events from leaves is required: A review of studies of urban plants used to reduce airborne particulate matter pollution. *Urban Forestry & Urban Greening*, *48*, 126559.

Appendix

In [ ]:

```python
import pandas as pd
import geopandas as gpd
import numpy as np

import glob

from shapely.ops import unary_union

import matplotlib.pyplot as plt
import seaborn as sns

from libpysal.weights import Kernel
from esda.moran import Moran

from scipy import stats
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import KFold, LeaveOneOut,
cross_val_predict
from sklearn.inspection import permutation_importance
```

Data cleaning

In [ ]:

```python
# read in all PM data
csv_files = glob.glob('data/AQMS' + '/*.csv')
df = pd.concat((pd.read_csv(f) for f in csv_files))
```

In [ ]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 402960 entries, 0 to 52559
Data columns (total 6 columns):
 #   Column                     Non-Null Count   Dtype  
---  ------                     --------------   -----  
 0   Site                       402960 non-null  object 
 1   Species                    402960 non-null  object 
 2   ReadingDateTime            402960 non-null  object 
 3   Value                      202670 non-null  float64
 4   Units                      402960 non-null  object 
 5   Provisional or Ratified    402960 non-null  object 
dtypes: float64(1), object(5)
memory usage: 21.5+ MB
```

In [ ]:

```python
# drop unnecessary columns
df.drop(['Species', 'Units', 'Provisional or Ratified'], axis=1,
```

```
inplace=True)
```

In [ ]:

```
df.groupby('Site').describe()
```

	Value							
	count	mean	std	min	25%	50%	75%	max
Site								
BL0	8558.0	10.750888	10.112520	-3.3	4.7	7.6	12.7	92.40000
BQ9	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
BT4	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
BX9	7169.0	11.813182	10.972091	-3.8	5.3	7.9	13.8	88.10000
BY7	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CD1	8544.0	11.132924	10.262592	-2.8	4.9	7.8	13.4	88.30000
CD9	8730.0	13.642887	10.411786	-7.3	7.2	10.9	16.3	83.90000
CE2	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
CR8	8711.0	10.115831	9.176507	-3.0	5.0	7.0	12.0	84.00000
CT2	8437.0	13.957568	10.865349	-3.0	8.0	11.0	16.0	441.00000
CT3	7575.0	11.669967	10.486332	-3.0	6.0	9.0	15.0	251.00000
GB0	8637.0	12.176705	9.036808	-1.2	6.7	9.4	14.1	79.80000
GN0	3193.0	11.319449	9.740894	-7.2	4.9	8.3	14.9	65.10000
GN3	8342.0	13.411832	11.277777	-3.5	6.8	9.6	15.5	109.40000
GN6	8252.0	10.966893	9.999743	-4.2	5.1	7.7	12.5	84.10000
GR4	8516.0	10.863269	9.913018	-2.7	5.2	8.0	12.5	97.60000
GR8	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
GR9	8713.0	10.425215	10.639660	-4.3	4.0	6.9	12.6	84.50000
HG1	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
HP1	8756.0	9.933029	9.987813	0.4	4.2	6.5	11.3	90.90000
HR1	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
HV1	8403.0	11.004820	12.916148	-9.0	4.5	7.8	13.5	472.20001
KC1	8723.0	9.579548	9.470490	0.4	4.2	6.3	11.0	121.00000
KF1	8723.0	9.578723	9.470523	0.4	4.1	6.4	11.0	121.00000
LH0	8510.0	9.538249	9.165456	0.4	4.1	6.3	11.2	91.00000
LW2	7742.0	14.953500	11.325410	-6.2	8.0	11.6	17.9	92.60000
LW5	411.0	8.671533	7.945340	-4.0	3.0	7.0	13.0	40.00000
MY7	7948.0	14.347974	10.963840	-3.3	7.5	11.4	17.5	93.00000
NM2	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	Value							
	count	mean	std	min	25%	50%	75%	max
Site								
NM3	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
RB7	3399.0	11.471315	10.159918	-4.0	6.0	9.0	15.0	272.00000
RD0	3249.0	8.211480	8.305129	-5.0	3.0	6.0	11.0	80.00000
SK6	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SK8	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SK9	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SKA	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SKB	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SKC	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ST5	8648.0	11.718316	10.163471	-7.0	6.0	9.0	14.0	99.00000
TD5	8148.0	11.784892	14.797966	0.0	5.8	8.6	13.4	592.79999
TH4	6478.0	13.380195	11.332006	-5.3	6.5	9.7	16.1	152.30000
TK3	7940.0	11.548237	11.593650	-2.0	5.0	8.0	14.0	111.00000
TK9	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
TL6	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
WM0	2215.0	11.681716	9.942168	0.0	7.0	9.0	14.0	339.00000
WMD	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

GN0, LW5, RB7, RD0, WM0 has too few data (less than half of the total amount)

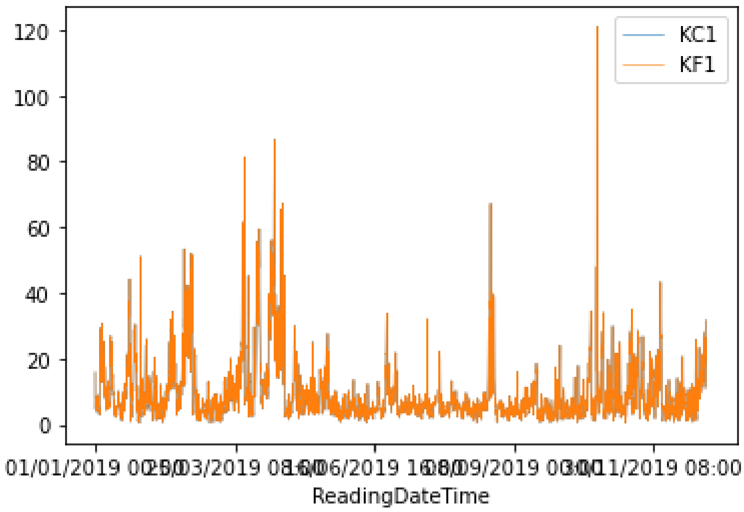
In [ ]:
```
# list of site codes with valid PM data
valid_AQMS = df.dropna()['Site'].unique().tolist()

# remove the site with few data from the list
for site in ['GN0', 'LW5', 'RB7', 'RD0', 'WM0']:
    valid_AQMS.remove(site)

# clean the PM dataset
df = df[df['Site'].isin(valid_AQMS)]
df = df.reset_index(drop=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 201480 entries, 0 to 201479
Data columns (total 3 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Site            201480 non-null  object
```

```
 1   ReadingDateTime  201480 non-null  object
 2   Value            190203 non-null  float64
dtypes: float64(1), object(2)
memory usage: 4.6+ MB
```

In [ ]:

```python
# KF1 and KC1 are very similar
fig,ax = plt.subplots()
df[df['Site']=='KC1'].plot(x='ReadingDateTime', y='Value', ax=ax,
label='KC1', linewidth=0.5)
df[df['Site']=='KF1'].plot(x='ReadingDateTime', y='Value', ax=ax,
label='KF1', linewidth=0.5)
plt.show()
```



In [ ]:

```python
# Remove KF1
df = df[df['Site']!='KF1']
valid_AQMS.remove('KF1')
```

In [ ]:

```python
len(df['Site'].unique())
```

```
22
```

In [ ]:

```python
df.groupby('Site').describe()
```

	Value							
	count	mean	std	min	25%	50%	75%	max
Site								
BL0	8558.0	10.750888	10.112520	-3.3	4.7	7.6	12.7	92.40000
BX9	7169.0	11.813182	10.972091	-3.8	5.3	7.9	13.8	88.10000
CD1	8544.0	11.132924	10.262592	-2.8	4.9	7.8	13.4	88.30000
CD9	8730.0	13.642887	10.411786	-7.3	7.2	10.9	16.3	83.90000
CR8	8711.0	10.115831	9.176507	-3.0	5.0	7.0	12.0	84.00000
CT2	8437.0	13.957568	10.865349	-3.0	8.0	11.0	16.0	441.00000

	Value							
	count	mean	std	min	25%	50%	75%	max
Site								
CT3	7575.0	11.669967	10.486332	-3.0	6.0	9.0	15.0	251.00000
GB0	8637.0	12.176705	9.036808	-1.2	6.7	9.4	14.1	79.80000
GN3	8342.0	13.411832	11.277777	-3.5	6.8	9.6	15.5	109.40000
GN6	8252.0	10.966893	9.999743	-4.2	5.1	7.7	12.5	84.10000
GR4	8516.0	10.863269	9.913018	-2.7	5.2	8.0	12.5	97.60000
GR9	8713.0	10.425215	10.639660	-4.3	4.0	6.9	12.6	84.50000
HP1	8756.0	9.933029	9.987813	0.4	4.2	6.5	11.3	90.90000
HV1	8403.0	11.004820	12.916148	-9.0	4.5	7.8	13.5	472.20001
KC1	8723.0	9.579548	9.470490	0.4	4.2	6.3	11.0	121.00000
LH0	8510.0	9.538249	9.165456	0.4	4.1	6.3	11.2	91.00000
LW2	7742.0	14.953500	11.325410	-6.2	8.0	11.6	17.9	92.60000
MY7	7948.0	14.347974	10.963840	-3.3	7.5	11.4	17.5	93.00000
ST5	8648.0	11.718316	10.163471	-7.0	6.0	9.0	14.0	99.00000
TD5	8148.0	11.784892	14.797966	0.0	5.8	8.6	13.4	592.79999
TH4	6478.0	13.380195	11.332006	-5.3	6.5	9.7	16.1	152.30000
TK3	7940.0	11.548237	11.593650	-2.0	5.0	8.0	14.0	111.00000

In [ ]:

```
# read in AQMS location geometry
gdf = gpd.read_file('data/AQMS/AQMS.gpkg')
gdf.head()
```

```
C:\Users\Yulun\anaconda3\envs\sds2021\lib\site-packages\geopandas\geodatafr
ame.py:577: RuntimeWarning: Sequential read of iterator was interrupted. Re
setting iterator. This can negatively impact the performance.
  for feature in features_lst:
```

	classification	dataowner	easting	latitude	longitude
0	Airport	None	542525.2800145757	51.5028	0.0521
1	Airport	None	542948.1357935619	51.5028	0.058193
2	Breathe London	None	535618.12376207381	51.521017999999998	-0.046672999999999999

	classification	dataowner	easting	latitude	longitude
3	Airport	None	542295.805364199	51.5074	0.049
4	Breathe London	None	524303.28797191242	51.604480000000002	-0.206490000000000001

5 rows × 21 columns

In [ ]:

```
# drop unnecessary columns
gdf = gdf.loc[:,['latitude', 'longitude', 'siteid', 'sitename']]
gdf.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 236 entries, 0 to 235
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   latitude   236 non-null    object
 1   longitude  236 non-null    object
 2   siteid     236 non-null    object
 3   sitename   236 non-null    object
dtypes: object(4)
memory usage: 7.5+ KB
```

In [ ]:

```
# check if all sites with data are within the geometry dataframe
for elem in valid_AQMS:
    if elem not in gdf['siteid'].unique().tolist():
        print(elem)
```

```
TK3
```

TK3: Thurrock - Stanford-le-Hope

```
51.518162000000, 0.4395480000000
```

Thurrock is not in London, so ignore

In [ ]:

```
# remove TK3 from the list and the dataframe
valid_AQMS.remove('TK3')
df = df[df['Site']≠'TK3']
```

In [ ]:

```
len(valid_AQMS)
```

```
21
```

In [ ]:

```
len(df['Site'].unique())
```

```
21
```

In [ ]:

```python
# get the geometry of the 21 sites
AQMS_gdf = gdf[gdf['siteid'].isin(valid_AQMS)]
AQMS_gdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21 entries, 115 to 232
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   latitude   21 non-null     object
 1   longitude  21 non-null     object
 2   siteid     21 non-null     object
 3   sitename   21 non-null     object
dtypes: object(4)
memory usage: 840.0+ bytes
```

In [ ]:

```python
# set to proper data tyeps
AQMS_gdf = AQMS_gdf.astype({'latitude':'float64',
'longitude':'float64',
                            'siteid':'string', 'sitename':'string'})
AQMS_gdf.dtypes
```

```
latitude     float64
longitude    float64
siteid        string
sitename      string
dtype: object
```

In [ ]:

```python
# generate geometry column based on lat and lon
AQMS_gdf = gpd.GeoDataFrame(AQMS_gdf,

geometry=gpd.points_from_xy(AQMS_gdf.longitude, AQMS_gdf.latitude),
                            crs='EPSG:4326')
```

In [ ]:

```python
# set the crs to british national grid
AQMS_gdf = AQMS_gdf.to_crs(27700)

# drop the lat and lon columns
AQMS_gdf = AQMS_gdf.drop(['latitude', 'longitude'], axis=1)
```

In [ ]:

```python
# save the geometry for future use
AQMS_gdf.to_file('data/AQMS_loc.shp')
```

In [ ]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 183960 entries, 0 to 201479
Data columns (total 3 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   Site             183960 non-null  object 
 1   ReadingDateTime  183960 non-null  object 
 2   Value            173540 non-null  float64
dtypes: float64(1), object(2)
memory usage: 5.6+ MB
```

In [ ]:

```
df['Value'].describe()
```

```
count    173540.000000
mean         11.721337
std          10.793000
min          -9.000000
25%           5.400000
50%           8.600000
75%          14.000000
max         592.799990
Name: Value, dtype: float64
```

There are many null values and negative values (not efficient because PM readings cannot be negative)

According to this, using mean-before-after is an approach.

In [ ]:

```
# set all negative reading to np.nan
# because you cannot have negative reading for PM concentration
df['Value'] = df['Value'].where(df['Value']>0, np.nan)
```

In [ ]:

```
val = df['Value'].values.copy()
```

In [ ]:

```
# check the number of null values
sum(val<=0), sum(np.isnan(val))
```

```
(0, 11847)
```

In [ ]:

```
# make sure that every site's first and last value is not null
for s in range(21):
    print(val[s*8760], val[s*8760-1])
```

```
13.0 31.3
17.0 29.0
36.0 33.0
21.4 35.0
16.1 30.0
29.1 23.6
23.3 30.0
53.3 38.9
43.1 31.2
14.7 22.6
```

```
15.8 34.5
11.2 31.7
14.0 33.9
26.3 37.0
22.0 34.8
35.1 33.0
20.4 28.6
20.5 32.9
11.0 30.6
18.2 33.9
35.5 24.1
```

In [ ]:

```python
# fill null values using mean-before-after method
for i in range(len(val)):
    if np.isnan(val[i]):
        j = 1
        while (np.isnan(val[i+j])) & (j < 12):
            j += 1

        # if there are 12 continous null values,
        # fill them with data for the same period for the previous day
        if j==12:
            for z in range(j+1):
                val[i+z] = val[i+z-24]
        val[i] = val[i-1] + (val[i+j] - val[i-1]) / (j+1)
```

In [ ]:

```python
sum(val<=0), sum(np.isnan(val))
```

```
(0, 0)
```

In [ ]:

```python
# cover the data in the df
df['Value'] = val
```

In [ ]:

```python
df.describe()
```

	Value
count	183960.000000
mean	11.775446
std	10.661761
min	0.100000
25%	5.500000
50%	8.600000
75%	14.000000
max	592.799990

In [ ]:

```
df.groupby('Site').describe()
```

	Value							
	count	mean	std	min	25%	50%	75%	max
Site								
BL0	8760.0	10.908521	10.228363	0.1	4.7	7.600000	12.800000	92.40000
BX9	8760.0	11.170749	10.396609	0.2	5.3	7.380917	12.600000	88.10000
CD1	8760.0	11.058464	10.162193	0.1	4.8	7.800000	13.300000	88.30000
CD9	8760.0	13.712563	10.330619	0.1	7.3	10.900000	16.300000	83.90000
CR8	8760.0	10.125421	9.129344	1.0	5.0	7.000000	12.000000	84.00000
CT2	8760.0	13.902287	10.708376	1.0	8.0	11.000000	16.000000	441.00000
CT3	8760.0	12.142583	10.057463	1.0	6.0	9.000000	16.000000	251.00000
GB0	8760.0	12.569166	9.864263	0.1	6.8	9.400000	14.325000	79.80000
GN3	8760.0	13.363480	11.089737	0.1	6.7	9.600000	15.700000	109.40000
GN6	8760.0	11.039737	9.829372	0.1	5.2	7.800000	12.700000	84.10000
GR4	8760.0	10.887037	9.764517	0.1	5.3	8.000000	12.600000	97.60000
GR9	8760.0	10.482015	10.585919	0.1	4.0	7.000000	12.600000	84.50000
HP1	8760.0	9.931490	9.985798	0.4	4.2	6.500000	11.300000	90.90000
HV1	8760.0	11.368690	12.671719	0.1	4.8	7.900000	13.600000	472.20001
KC1	8760.0	9.567551	9.452367	0.4	4.2	6.400000	11.000000	121.00000
LH0	8760.0	9.412646	9.069698	0.4	4.1	6.300000	10.925000	91.00000
LW2	8760.0	15.422345	11.470787	0.3	8.4	12.000000	18.200000	92.60000
MY7	8760.0	14.190663	10.758123	0.1	7.3	11.400000	17.400000	93.00000
ST5	8760.0	11.732403	10.095116	1.0	6.0	9.000000	14.000000	99.00000
TD5	8760.0	11.686217	14.387273	0.1	5.9	8.600000	13.400000	592.79999
TH4	8760.0	12.610338	10.015622	0.1	6.9	9.700000	14.561054	152.30000

In [ ]:

```
# save for future use
df.to_csv('data/hourly.csv', index=False)
```

Load in data

In [ ]:

```
# set seaborn theme
sns.set_theme(style='darkgrid')
```

In [ ]:

```
# read in AQMS locations
loc_gdf = gpd.read_file('data/AQMS_loc.shp')
```

```python
# read in PM2.5 hourly data
dep_df = pd.read_csv('data/hourly.csv')
```

In [ ]:
```python
# set buffer zones around each site (1km)
loc_gdf['buffer_1km'] = loc_gdf['geometry'].buffer(1000)
```

road modify

In [ ]:
```python
LD_wards = gpd.read_file("data/LD_boundary/London-wards-
2018_ESRI/London_Ward_CityMerged.shp")
london = LD_wards.unary_union
london_gdf = gpd.GeoSeries(london)
london_gdf.to_file('data/london_boundary.shp')
```

In [ ]:
```python
for typ in ['RoadLink', 'RoadNode', 'MotorwayJunction']:
    exec("%s = gpd.GeoDataFrame()"%typ)
    for tile in ['SP_','SU_','TL_','TQ_']:
        path = "data/oproad_essh_gb/data/%s%s.shp"%(tile, typ)
        exec("%s%s = gpd.read_file(path)"%(tile, typ))
        exec("%s = %s.append(%s%s, ignore_index=True)"%(typ, typ, tile,
typ))
```

In [ ]:
```python
spatial_index = RoadLink.sindex
bbox = london.bounds
sidx = list(spatial_index.intersection(bbox))
RoadLink_sub = RoadLink.iloc[sidx]

RoadLink_clip = RoadLink_sub.copy()
RoadLink_clip['geometry'] = RoadLink_sub.intersection(london)
```

In [ ]:
```python
RoadLink_clip = RoadLink_clip.reset_index(drop=True)
Rd = RoadLink_clip[RoadLink_clip['geometry'] != RoadLink_clip.loc[0,
'geometry']].reset_index(drop=True)
Rd.head()
```

In [ ]:
```python
Rd.to_file('data/london_Road.shp')
```

gsp modify

In [ ]:

```
# for downloading greenspace geometry
buffer_gdf = loc_gdf[['buffer_1km']]
buffer_gdf = gpd.GeoDataFrame(buffer_gdf, geometry='buffer_1km')
buffer_gdf.to_file('data/buffer.shp')
```

In [ ]:

```
loc_gdf
```

0575 - LH0

1065, 1070, 1565, 1570 - TD5

2565, 2570, 3065, 3070 - CR8

2565, 3065 - ST5

2080, 2580 - KC1

2580, 2585 - CD1

2580 - MY7

2580, 3080 - BL0, CD9

3080 - CT2, CT3

3570, 3575 - HP1, LW2

3575, 3580, 4075, 4080 - GN6

3580 - TH4

4070, 4075, 4570, 4575 - GB0

4070, 4075 - GR9, GR4

4075, 4575 - GN3

5075 - BX9

5080 - HV1

In [ ]:

```
def readin_Gsp(file_name, path='data/OSMM Greenspaces/tq/TQ',
suffix='_GreenspaceArea.shp'):
    if type(file_name) == str:
        gdf = gpd.read_file(path+file_name+suffix)
    else:
        gdf = pd.concat(gpd.read_file(path+f+suffix) for f in
file_name)
    return gdf
```

In [ ]:

```
loc_gdf['Gsp'] = gpd.GeoSeries()
```

```
In [ ]: loc_gdf.columns.get_loc('Gsp')
```

```
In [ ]: def get_Gsp(file_name, index):
            gdf = readin_Gsp(file_name)
            print('Finish reading in shapefile(s)')
            shp = gdf['geometry'].unary_union
            print('Finish unary union.')
            if type(index) == int:
                loc_gdf.iat[index, 4] = shp.intersection(loc_gdf.loc[index,
        'buffer_1km'])
            elif type(index) == list:
                for i in index:
                    loc_gdf.iat[i, 4] = shp.intersection(loc_gdf.loc[i,
        'buffer_1km'])
            else:
                print('invalid type!')
```

```
In [ ]: get_Gsp('0575', 13)
```

```
In [ ]: get_Gsp(['1065','1070','1565','1570'], 17)
```

```
In [ ]: get_Gsp(['2565','2570','3065','3070'], 6)
```

```
In [ ]: get_Gsp(['2565','3065'], 18)
```

```
In [ ]: get_Gsp(['2080','2580'], 14)
```

```
In [ ]: get_Gsp(['2580','2585'], 3)
```

```
In [ ]: get_Gsp('2580', 20)
```

```
In [ ]: get_Gsp(['2580','3080'], [1,2])
```

```
In [ ]: get_Gsp('3080', [4,5])
```

```
In [ ]: get_Gsp(['3570','3575'], [15,16])
```

```
In [ ]: get_Gsp(['3575','3580','4075','4080'], 9)
```

In [ ]:
```python
get_Gsp('3580', 19)
```

In [ ]:
```python
get_Gsp(['4070','4075','4570','4575'], 8)
```

In [ ]:
```python
get_Gsp(['4075', '4575'], [7, 11])
```

In [ ]:
```python
get_Gsp(['4075', '4575'], 10)
```

In [ ]:
```python
get_Gsp('5075', 0)
```

In [ ]:
```python
get_Gsp('5080', 12)
```

In [ ]:
```python
Gsp_gdf = loc_gdf[['siteid','Gsp']]
Gsp_gdf = Gsp_gdf.set_geometry('Gsp')
Gsp_gdf = Gsp_gdf.set_crs(27700)
Gsp_gdf.crs
```

In [ ]:
```python
Gsp_gdf.to_file('data/gsp_buffer_1km.shp')
```

generate near-road gsp

In [ ]:
```python
# read in (modified) greenspace geometry
Gsp_gdf = gpd.read_file('data/gsp_buffer_1km.shp')
```

In [ ]:
```python
# add the gsp geometry column to the location gdf
loc_gdf['Gsp'] = Gsp_gdf['geometry']
loc_gdf.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   siteid      21 non-null     object
 1   sitename    21 non-null     object
 2   geometry    21 non-null     geometry
 3   buffer_1km  21 non-null     geometry
 4   Gsp         21 non-null     geometry
dtypes: geometry(3), object(2)
memory usage: 968.0+ bytes
```

In [ ]:
```python
# save memory
del Gsp_gdf
```

In [ ]:

```
# Read in road (modified) geometry
Rd_gdf = gpd.read_file('data/london_Road.shp')
Rd_gdf.head()
```

Out[ ]:

	fictitious	identifier	class	roadNumber	name1	name1_lang	name2	name:
0	false	8CC0934A-4A4A-435A-BEBB-521AD3E8C143	Not Classified	None	The Bridlepath	None	None	
1	false	ECE86DA8-118A-46AB-8D5D-56F68B96E7BB	Unclassified	None	Ditches Lane	None	None	
2	false	960A1B1E-15CD-4E9C-816C-4F79CB0442E7	A Road	A233	Main Road	None	None	
3	false	0E0182BB-7E46-4250-B9EE-37D58BA0E73C	Unclassified	None	Grays Road	None	None	
4	false	A6456BD8-2D7F-4CE9-9192-112965FA7AD1	Unclassified	None	Old Fox Close	None	None	

In [ ]:

```
for c in Rd_gdf['class'].unique():
    print('Number of ' + c + ': ', Rd_gdf[Rd_gdf['class'] ==
c].shape[0])
```

```
Number of Not Classified:  14061
Number of Unclassified:  117392
Number of A Road:  25452
Number of B Road:  6734
Number of Unknown:  36448
Number of Classified Unnumbered:  8925
Number of Motorway:  189
```

In [ ]:

```
# Get all types of roads
Rd = {}
for c in Rd_gdf['class'].unique():
    Rd[c] = Rd_gdf[Rd_gdf['class'] == c].loc[:, 'geometry'].unary_union
Rd
```

Out[ ]:

```
{'Not Classified': <shapely.geometry.multilinestring.MultiLineString at 0x1
ef30405670>,
 'Unclassified': <shapely.geometry.multilinestring.MultiLineString at 0x1ef
275d6e80>,
 'A Road': <shapely.geometry.multilinestring.MultiLineString at 0x1ef304055
```

```
e0>,
 'B Road': <shapely.geometry.multilinestring.MultiLineString at 0x1ef276997
60>,
 'Unknown': <shapely.geometry.multilinestring.MultiLineString at 0x1ef302cc
0d0>,
 'Classified Unnumbered': <shapely.geometry.multilinestring.MultiLineString
at 0x1ef275d6dc0>,
 'Motorway': <shapely.geometry.multilinestring.MultiLineString at 0x1ef3040
5580>}
```

In [ ]:

```python
# merge Not Classified and Unknown into one category
Rd['Other'] = unary_union([Rd['Not Classified'], Rd['Unknown']])
Rd.pop('Not Classified')
Rd.pop('Unknown')
Rd
```

Out[ ]:

```
{'Unclassified': <shapely.geometry.multilinestring.MultiLineString at 0x1ef
275d6e80>,
 'A Road': <shapely.geometry.multilinestring.MultiLineString at 0x1ef304055
e0>,
 'B Road': <shapely.geometry.multilinestring.MultiLineString at 0x1ef276997
60>,
 'Classified Unnumbered': <shapely.geometry.multilinestring.MultiLineString
at 0x1ef275d6dc0>,
 'Motorway': <shapely.geometry.multilinestring.MultiLineString at 0x1ef3040
5580>,
 'Other': <shapely.geometry.multilinestring.MultiLineString at 0x1ef275c322
0>}
```

In [ ]:

```python
del Rd_gdf
```

In [ ]:

```python
# add the road geometries to the location gdf
for key in Rd.keys():
    loc_gdf[key] = loc_gdf['buffer_1km'].intersection(Rd[key])
loc_gdf.head()
```

Out[ ]:

	siteid	sitename	geometry	buffer_1km	Gsp	Unclassified	A
0	BX9	Bexley - Slade Green FDMS	POINT (551862.205 176375.976)	POLYGON ((552862.205 176375.976, 552857.390 17...	MULTIPOLYGON Z (((551468.680 175909.000 0.000,...	MULTILINESTRING Z ((552075.170 175434.690 0.00...	MULTILINES Z ((5524 175621.08(
1	BL0	Camden - Bloomsbury	POINT (530120.048 182038.807)	POLYGON ((531120.048 182038.807, 531115.233 18...	MULTIPOLYGON Z (((530046.600 181557.850 0.000,...	MULTILINESTRING Z ((530175.051 181041.510 0.00...	MULTILINES Z ((5299 181058.68(

	siteid	sitename	geometry	buffer_1km	Gsp	Unclassified	[illegible]
2	CD9	Camden - Euston Road	POINT (529900.870 182666.124)	POLYGON ((530900.870 182666.124, 530896.055 18...	MULTIPOLYGON Z (((530164.744 181702.568 0.000,...	MULTILINESTRING Z ((529650.665 181699.144 0.00...	MULTILINES[illegible] Z ((5299[illegible] 181667.10[illegible]
3	CD1	Camden - Swiss Cottage	POINT (526629.730 184391.024)	POLYGON ((527629.730 184391.024, 527624.915 18...	MULTIPOLYGON Z (((527127.744 183525.057 0.000,...	MULTILINESTRING Z ((526949.610 183444.673 0.00...	MULTILINES[illegible] Z ((5266[illegible] 183408.05[illegible]
4	CT2	City of London - Farringdon Street	POINT (531622.273 181213.818)	POLYGON ((532622.273 181213.818, 532617.458 18...	MULTIPOLYGON Z (((532257.200 181585.050 0.000,...	MULTILINESTRING Z ((531742.953 180221.995 0.00...	MULTILINES[illegible] Z ((5316[illegible] 180215.42[illegible]

In [ ]:

```
del Rd
```

In [ ]:

```
# Rename the columns
loc_gdf.rename(columns={'Unclassified': 'UnC',
                        'A Road': 'A',
                        'B Road': 'B',
                        'Classified Unnumbered': 'CUn',
                        'Motorway': 'Mt'}, inplace=True)

# save the road classification to a list
Rd_type = loc_gdf.columns[-6:].tolist()
Rd_type
```

Out[ ]:

```
['UnC', 'A', 'B', 'CUn', 'Mt', 'Other']
```

In [ ]:

```
# Get all near-road greenspaces
for col in Rd_type:
    loc_gdf['n'+col+'_Gsp'] =
loc_gdf['Gsp'].intersection(loc_gdf[col].buffer(50))

loc_gdf.head()
```

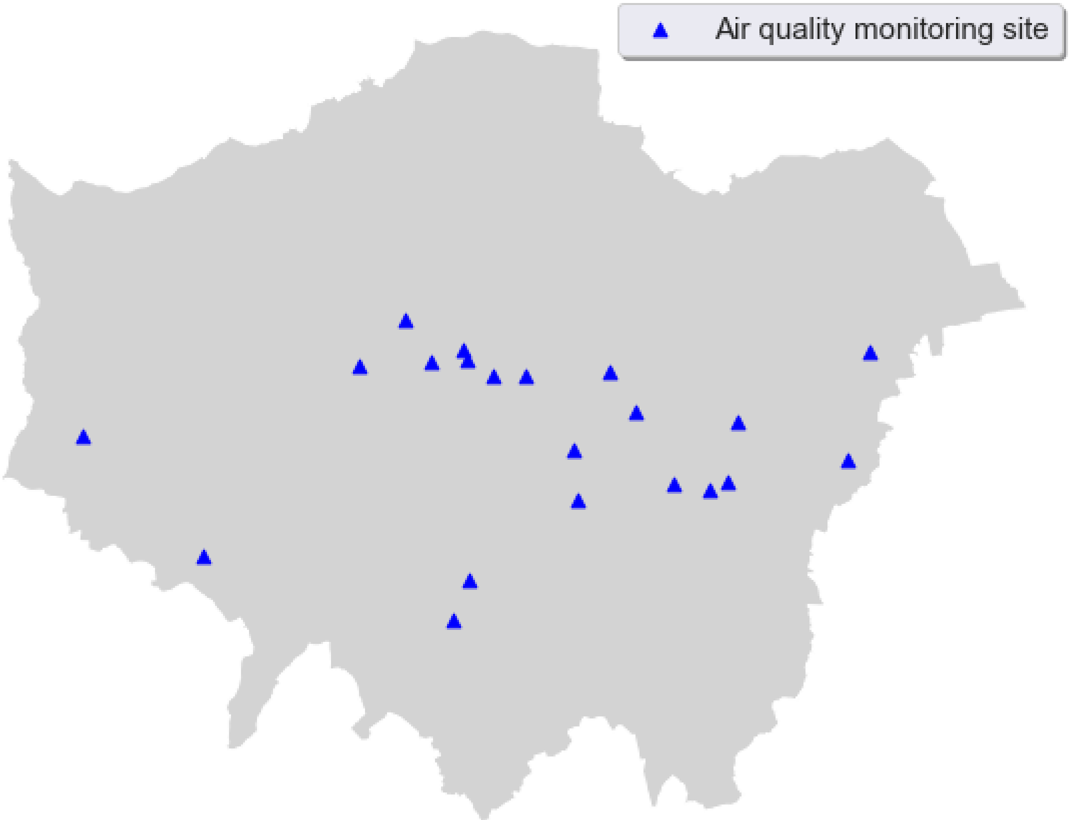
Out[ ]:

	siteid	sitename	geometry	buffer_1km	Gsp	UnC	
0	BX9	Bexley - Slade Green FDMS	POINT (551862.205 176375.976)	POLYGON ((552862.205 176375.976, 552857.390 17...	MULTIPOLYGON Z (((551468.680 175909.000 0.000,...	MULTILINESTRING Z ((552075.170 175434.690 0.00...	MULTILINES[illegible] Z ((5524[illegible] 175621.08[illegible]

	siteid	sitename	geometry	buffer_1km	Gsp	UnC	
1	BL0	Camden - Bloomsbury	POINT (530120.048 182038.807)	POLYGON ((531120.048 182038.807, 531115.233 18...	MULTIPOLYGON Z (((530046.600 181557.850 0.000,...	MULTILINESTRING Z ((530175.051 181041.510 0.00...	MULTILINES Z ((5299 181058.68(
2	CD9	Camden - Euston Road	POINT (529900.870 182666.124)	POLYGON ((530900.870 182666.124, 530896.055 18...	MULTIPOLYGON Z (((530164.744 181702.568 0.000,...	MULTILINESTRING Z ((529650.665 181699.144 0.00...	MULTILINES Z ((5299 181667.10:
3	CD1	Camden - Swiss Cottage	POINT (526629.730 184391.024)	POLYGON ((527629.730 184391.024, 527624.915 18...	MULTIPOLYGON Z (((527127.744 183525.057 0.000,...	MULTILINESTRING Z ((526949.610 183444.673 0.00...	MULTILINES Z ((5266 183408.05(
4	CT2	City of London - Farringdon Street	POINT (531622.273 181213.818)	POLYGON ((532622.273 181213.818, 532617.458 18...	MULTIPOLYGON Z (((532257.200 181585.050 0.000,...	MULTILINESTRING Z ((531742.953 180221.995 0.00...	MULTILINES Z ((5316 180215.42:

In [ ]:

```
# Fig 1
fig,ax = plt.subplots(1, figsize=(12,8))

london_gdf.plot(color='lightgrey', ax=ax)

loc_gdf['geometry'].plot(markersize=45, marker='^', color='blue',
                         label='Air quality monitoring site', ax=ax)

ax.axis('off')

legend = ax.legend(loc='best', shadow=True, fontsize=15)

plt.savefig('figure/Fig1.png', facecolor=None, dpi=500)

plt.show()
```



There are some sites that seem to be very close to each other.

In [ ]:
```python
# add a column that specifies the shortest distance of a site to its nearest neighbour
loc_gdf['min_dis'] = pd.Series(dtype='float64')
for index, row in loc_gdf.iterrows():
    dis = []
    for i, v in loc_gdf['geometry'].iteritems():
        dis.append(row['geometry'].distance(v))
    dis.remove(0)
    loc_gdf.loc[index, 'min_dis'] = min(dis)
```

In [ ]:
```python
# list sites that are close to each other (within 1.5km)
loc_gdf[loc_gdf['min_dis'] <= 1500]
```

Out[ ]:

	siteid	sitename	geometry	buffer_1km	Gsp	UnC	
1	BL0	Camden - Bloomsbury	POINT (530120.048 182038.807)	POLYGON ((531120.048 182038.807, 531115.233 18...	MULTIPOLYGON Z (((530046.600 181557.850 0.000,...	MULTILINESTRING Z ((530175.051 181041.510 0.00...	MULTILINES Z ((5299 181058.68(
2	CD9	Camden - Euston Road	POINT (529900.870 182666.124)	POLYGON ((530900.870 182666.124, 530896.055 18...	MULTIPOLYGON Z (((530164.744 181702.568 0.000,...	MULTILINESTRING Z ((529650.665 181699.144 0.00...	MULTILINES Z ((5299 181667.10:

	siteid	sitename	geometry	buffer_1km	Gsp	UnC
7	GR4	Greenwich - Eltham	POINT (543978.694 174655.234)	POLYGON ((544978.694 174655.234, 544973.878 17...	MULTIPOLYGON Z (((544807.871 175213.894 0.000,...	MULTILINESTRING Z ((543437.000 173984.000 0.00...	MULTILINES Z ((5434 173917.89(
8	GB0	Greenwich - Falconwood FDMS	POINT (544997.933 175098.152)	POLYGON ((545997.933 175098.152, 545993.118 17...	MULTIPOLYGON Z (((544142.814 174582.038 0.000,...	MULTILINESTRING Z ((544952.089 174100.404 0.00...	MULTILINES Z ((5447 174454.12(

In [ ]:

```
# check their readings' descriptive statistics
dep_df[dep_df['Site'].isin(['BL0', 'CD9', 'GR4',
'GB0'])].groupby('Site').describe()
```

Out[ ]:

	Value							
	count	mean	std	min	25%	50%	75%	max
Site								
BL0	8760.0	10.908521	10.228363	0.1	4.7	7.6	12.800	92.4
CD9	8760.0	13.712563	10.330619	0.1	7.3	10.9	16.300	83.9
GB0	8760.0	12.569166	9.864263	0.1	6.8	9.4	14.325	79.8
GR4	8760.0	10.887037	9.764517	0.1	5.3	8.0	12.600	97.6

In [ ]:

```
# student's t test
stats.ttest_rel(dep_df[dep_df['Site']=='BL0'].Value.values,
                dep_df[dep_df['Site']=='CD9'].Value.values)
```

Out[ ]:

```
Ttest_relResult(statistic=-59.89747540590601, pvalue=0.0)
```

In [ ]:

```
stats.ttest_rel(dep_df[dep_df['Site']=='GR4'].Value.values,
                dep_df[dep_df['Site']=='GB0'].Value.values)
```

Out[ ]:

```
Ttest_relResult(statistic=-31.347923748114297, pvalue=1.5260626870045138e-2
04)
```

Both indicate that we should reject H0, meaning the two datasets are statistically significantly different.

In [ ]:

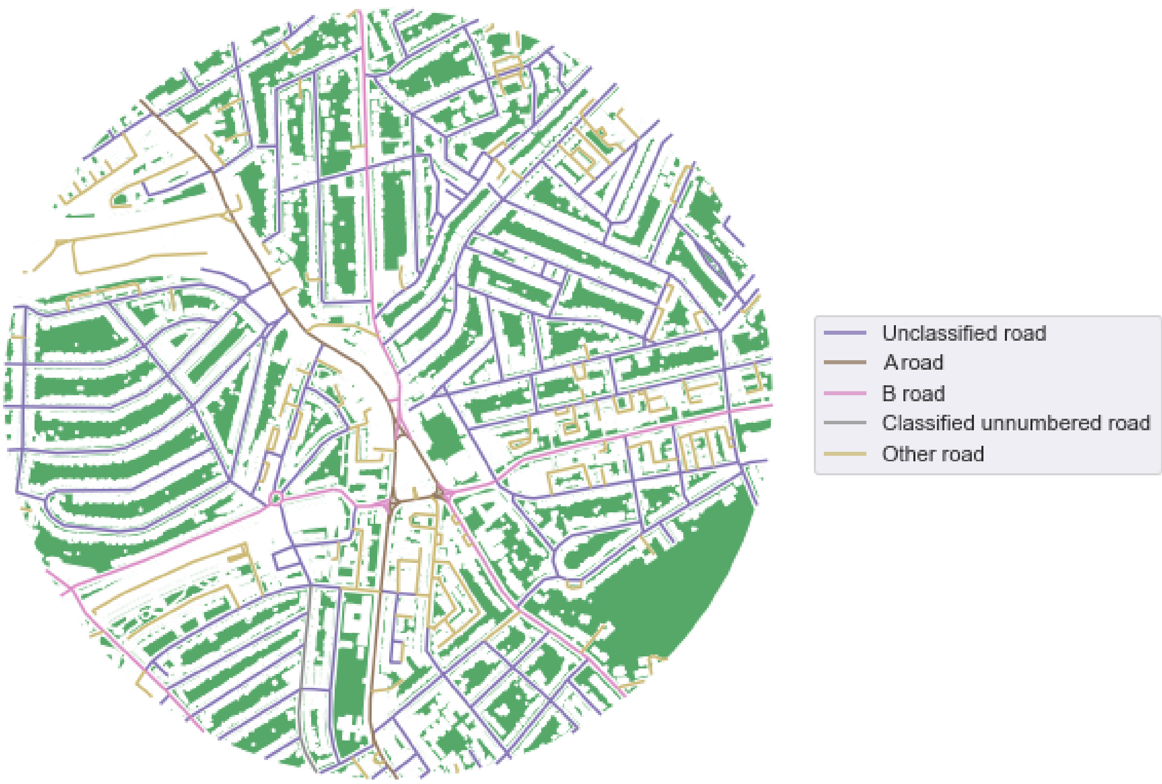
```
sns.color_palette()
```

Out[ ]:

In [ ]:

```
# Fig 4 - example of a site buffer (CD1 Camden-Swiss Cottage)
```

```
fig, ax = plt.subplots(1, figsize=(12,8))
loc_gdf.loc[[3],'buffer_1km'].plot(color='white', edgecolor=None,
ax=ax)
loc_gdf.loc[[3],'Gsp'].plot(color=sns.color_palette()[2],
edgecolor=None, ax=ax, label='Greenspace')
loc_gdf.loc[[3],'UnC'].plot(color=sns.color_palette()[4],
edgecolor=None, ax=ax, label='Unclassified road')
loc_gdf.loc[[3],'A'].plot(color=sns.color_palette()[5], edgecolor=None,
ax=ax, label='A road')
loc_gdf.loc[[3],'B'].plot(color=sns.color_palette()[6], edgecolor=None,
ax=ax, label='B road')
loc_gdf.loc[[3],'CUn'].plot(color=sns.color_palette()[7],
edgecolor=None, ax=ax, label='Classified unnumbered road')
loc_gdf.loc[[3],'Other'].plot(color=sns.color_palette()[8],
edgecolor=None, ax=ax, label='Other road')

plt.legend(bbox_to_anchor=(0.99,0.5), loc='center left')
ax.axis('off')

plt.savefig('figure/Fig4.png', facecolor=None, dpi=500)
plt.show()
```



In [ ]:

```
# get total areas of greenspaces
loc_gdf['Gsp_area'] = loc_gdf['Gsp'].area
```

In [ ]:
```python
# get road lengths of each type and near-road greenspaces for each type
for col in Rd_type:
    loc_gdf[col+'_len'] = loc_gdf[col].length
    loc_gdf[col+'_area_per_len'] = loc_gdf['n'+col+'_Gsp'].area /
loc_gdf[col+'_len']
```

In [ ]:
```python
loc_gdf.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 31 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   siteid               21 non-null     object
 1   sitename             21 non-null     object
 2   geometry             21 non-null     geometry
 3   buffer_1km           21 non-null     geometry
 4   Gsp                  21 non-null     geometry
 5   UnC                  21 non-null     geometry
 6   A                    21 non-null     geometry
 7   B                    21 non-null     geometry
 8   CUn                  21 non-null     geometry
 9   Mt                   21 non-null     geometry
 10  Other                21 non-null     geometry
 11  nUnC_Gsp             21 non-null     geometry
 12  nA_Gsp               21 non-null     geometry
 13  nB_Gsp               21 non-null     geometry
 14  nCUn_Gsp             21 non-null     geometry
 15  nMt_Gsp              21 non-null     geometry
 16  nOther_Gsp           21 non-null     geometry
 17  min_dis              21 non-null     float64
 18  Gsp_area             21 non-null     float64
 19  UnC_len              21 non-null     float64
 20  UnC_area_per_len     21 non-null     float64
 21  A_len                21 non-null     float64
 22  A_area_per_len       21 non-null     float64
 23  B_len                21 non-null     float64
 24  B_area_per_len       17 non-null     float64
 25  CUn_len              21 non-null     float64
 26  CUn_area_per_len     18 non-null     float64
 27  Mt_len               21 non-null     float64
 28  Mt_area_per_len      1 non-null      float64
 29  Other_len            21 non-null     float64
 30  Other_area_per_len   21 non-null     float64
dtypes: float64(14), geometry(15), object(2)
memory usage: 5.2+ KB
```

In [ ]:
```python
exp_df = loc_gdf.loc[:,['siteid']+[col+'_area_per_len' for col in
Rd_type]].copy()
exp_df.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   siteid              21 non-null     object 
 1   UnC_area_per_len    21 non-null     float64
 2   A_area_per_len      21 non-null     float64
 3   B_area_per_len      17 non-null     float64
 4   CUn_area_per_len    18 non-null     float64
 5   Mt_area_per_len     1 non-null      float64
 6   Other_area_per_len  21 non-null     float64
dtypes: float64(6), object(1)
memory usage: 1.3+ KB
```

There are many null values in `Mt_area_per_len` .

Because only one site has near motorway.

Remove the variable would be the best.

In [ ]:

```
exp_df.drop('Mt_area_per_len', axis=1, inplace=True)
loc_gdf.drop(['Mt_len', 'Mt_area_per_len'], axis=1, inplace=True)
Rd_type.remove('Mt')
```

Some null values in `B_area_per_len` and `CUn_area_per_len` , which is due to the lengths of B roads or Classified Unnumbered roads in these buffers are zero.

In [ ]:

```
# set the null values to zero
exp_df.fillna(0, inplace=True)
exp_df.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   siteid              21 non-null     object 
 1   UnC_area_per_len    21 non-null     float64
 2   A_area_per_len      21 non-null     float64
 3   B_area_per_len      21 non-null     float64
 4   CUn_area_per_len    21 non-null     float64
 5   Other_area_per_len  21 non-null     float64
dtypes: float64(5), object(1)
memory usage: 1.1+ KB
```

In [ ]:

```
loc_gdf.fillna(0, inplace=True)
```

In [ ]:

```
exp_df.to_csv('exp_data.csv', index=False)
```

Data analysis

In [ ]:
```
exp_df = pd.read_csv('exp_data.csv')
exp_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   siteid              21 non-null     object
 1   UnC_area_per_len    21 non-null     float64
 2   A_area_per_len      21 non-null     float64
 3   B_area_per_len      21 non-null     float64
 4   CUn_area_per_len    21 non-null     float64
 5   Other_area_per_len  21 non-null     float64
dtypes: float64(5), object(1)
memory usage: 1.1+ KB
```

In [ ]:
```
dep_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183960 entries, 0 to 183959
Data columns (total 3 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Site             183960 non-null  object
 1   ReadingDateTime  183960 non-null  object
 2   Value            183960 non-null  float64
dtypes: float64(1), object(2)
memory usage: 4.2+ MB
```

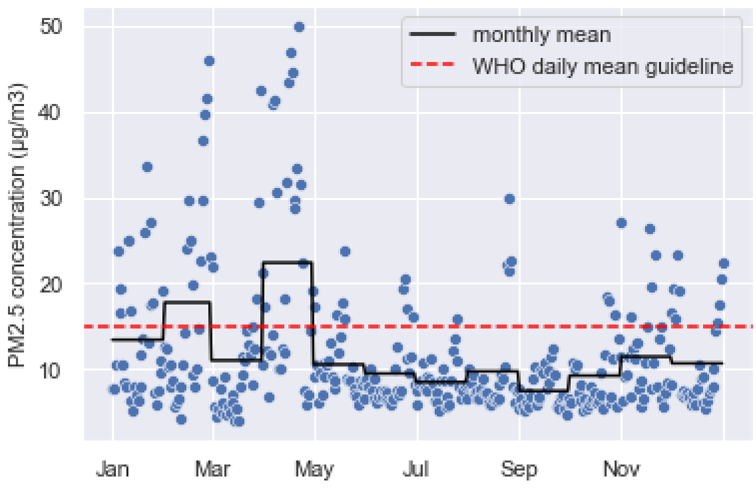
In [ ]:
```
# covert the DateTime column to numpy.datetime variable
dep_df['ReadingDateTime'] = pd.to_datetime(dep_df['ReadingDateTime'],
format="%d/%m/%Y %H:%M")
dep_df.rename(columns={'ReadingDateTime':'DateTime'}, inplace=True)
dep_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183960 entries, 0 to 183959
Data columns (total 3 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   Site      183960 non-null  object
 1   DateTime  183960 non-null  datetime64[ns]
 2   Value     183960 non-null  float64
dtypes: datetime64[ns](1), float64(1), object(1)
memory usage: 4.2+ MB
```

In [ ]:
```
dep_df['month'] = dep_df['DateTime'].dt.month
dep_df['hour'] = dep_df['DateTime'].dt.hour
dep_df['dayofmonth'] = dep_df['DateTime'].dt.day
dep_df['Date'] = dep_df['DateTime'].dt.date
```

In [ ]:

```
# Fig 2
mlabels =
['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

sns.scatterplot(x=dep_df['Date'].unique(),
y=dep_df.groupby('Date').mean()['Value'])

plt.plot(dep_df['Date'].unique(),

dep_df.groupby('Date').mean().merge(dep_df.groupby('month').mean()
[['Value']], left_on='month', right_index=True)['Value_y'],
         color='black', label='monthly mean')

plt.axhline(y=15, color='red', linestyle='--', label='WHO daily mean
guideline')

plt.ylabel('PM2.5 concentration (μg/m3)', fontsize=11)
plt.gca().set_xticks(plt.gca().get_xticks())
plt.gca().set_xticklabels([mlabels[2*i] for i in range(6)]+[''])

plt.legend()

plt.savefig('figure/Fig2.png', facecolor=None, dpi=500)

plt.show()
```



In [ ]:

```
# number of date above WHO guideline
(dep_df.groupby('Date').mean()['Value']>15).sum()
```

Out[ ]: 74

In [ ]:

```
# annual mean for each site - table 1
```

```
dep_df.groupby('Site').mean()['Value']
```

Out[ ]:

```
Site
BL0    10.908521
BX9    11.170749
CD1    11.058464
CD9    13.712563
CR8    10.125421
CT2    13.902287
CT3    12.142583
GB0    12.569166
GN3    13.363480
GN6    11.039737
GR4    10.887037
GR9    10.482015
HP1     9.931490
HV1    11.368690
KC1     9.567551
LH0     9.412646
LW2    15.422345
MY7    14.190663
ST5    11.732403
TD5    11.686217
TH4    12.610338
Name: Value, dtype: float64
```

In [ ]:

```
# annual mean for London
dep_df['Value'].mean()
```

Out[ ]:

```
11.775446103783608
```

In [ ]:

```
# explanatory variable names to a list
var_names = exp_df.columns[1:].tolist()
var_names
```

Out[ ]:

```
['UnC_area_per_len',
 'A_area_per_len',
 'B_area_per_len',
 'CUn_area_per_len',
 'Other_area_per_len']
```

In [ ]:

```
# Gaussian kernel weights matrix
weight = Kernel.from_dataframe(loc_gdf, geom_col='geometry',
function='gaussian')
```

In [ ]:

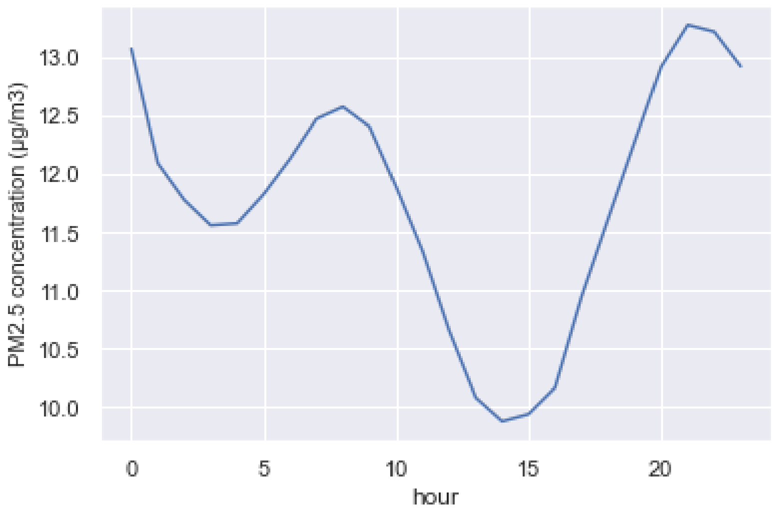
```
# check global moran's I for the explanatory variables
for var in var_names:
    moran_temp = Moran(exp_df[var].values, weight)
    print("Global Moran's I for " + var + ' is ', round(moran_temp.I,
```

```
5),
          ' p-value: ', round(moran_temp.p_norm, 5))
```

```
Global Moran's I for UnC_area_per_len is  0.22651  p-value:  0.0
Global Moran's I for A_area_per_len is  0.1898  p-value:  4e-05
Global Moran's I for B_area_per_len is  0.02886  p-value:  0.17844
Global Moran's I for CUn_area_per_len is  0.15787  p-value:  0.00039
Global Moran's I for Other_area_per_len is  0.19556  p-value:  3e-05
```

In [ ]:

```
# Fig 3 - annual mean per hour
dep_df.groupby('hour').mean()['Value'].plot()
plt.ylabel('PM2.5 concentration (μg/m3)', fontsize=11)

plt.savefig('figure/Fig3.png', facecolor=None, dpi=500)

plt.show()
```



In [ ]:

```
# table 2
exp_df[var_names].describe()
```

Out[ ]:

	UnC_area_per_len	A_area_per_len	B_area_per_len	CUn_area_per_len	Other_area_per_len
count	21.000000	21.000000	21.000000	21.000000	21.00000C
mean	28.520807	26.246194	25.788827	34.007412	42.49722€
std	14.016818	16.163453	21.169168	22.542926	21.02656€
min	6.461089	7.084152	0.000000	0.000000	16.415021
25%	13.888893	12.595197	5.633153	17.717612	26.381062
50%	32.957256	21.628544	27.486036	33.960430	37.11636ç
75%	38.753776	35.940536	44.131224	50.730381	55.76674ç
max	47.129151	60.890307	56.162862	64.512992	84.75567ç

In [ ]:

```
# feature importance function
```

```
def get_importance(reg, features, target, feature_names, state=25,
rep=50, method='r2'):
    mean = []
    std = []
    reg.fit(features, target)
    importance = permutation_importance(reg, features, target,
n_repeats=rep,
                                        random_state=state,
scoring=method)
    for i in range(len(feature_names)):
        mean.append(round(importance.importances_mean[i], 5))
        std.append(round(importance.importances_std[i], 5))
    return mean, std
```

In [ ]:

```
# cross-validation function
def get_cv(reg, features, target, iter=100, n_splits=5, loo=False):
    cv_r2 = []
    cv_resid = []
    if loo:
        split = LeaveOneOut()
        iter = 1
    for i in range(iter):
        if not loo:
            split = KFold(n_splits=n_splits, shuffle=True,
random_state=i)
        cvprd = cross_val_predict(reg, features, target, cv=split)

        r = stats.pearsonr(target,cvprd)[0]
        resid = cvprd - target

        cv_r2.append(r**2)
        cv_resid.append(resid)

    return [round(np.mean(cv_r2),5), round(np.std(cv_r2),5),
            np.mean(np.array(cv_resid), axis=0)]
```

In [ ]:

```
# initialise linear model
reg = LinearRegression()
```

In [ ]:

```
# df for annual mean
annual = exp_df.merge(dep_df.groupby('Site').mean()[['Value']],
```

```python
left_on='siteid', right_index=True)
annual.head()
```

Out[ ]:

	siteid	UnC_area_per_len	A_area_per_len	B_area_per_len	CUn_area_per_len	Other_area_per_
0	BX9	42.547777	34.722332	0.000000	63.364634	52.384
1	BL0	10.218919	9.464790	16.140991	0.000000	21.815
2	CD9	13.888893	12.595197	20.121072	0.000000	24.772
3	CD1	33.768627	16.790598	32.863003	49.419568	37.116
4	CT2	6.777993	7.084152	5.633153	30.535483	17.630

In [ ]:

```python
# global moran's I for annual mean
Moran(annual['Value'].values, weight).I
```

Out[ ]:

```
0.0960805356530469
```

In [ ]:

```python
# model variables
y = annual['Value'].values
X = annual[var_names].values
```

In [ ]:

```python
# feature importance for annual mean model
get_importance(reg, X, y, var_names)
```

Out[ ]:

```
([1.63613, 0.19492, 0.49033, 0.61434, 0.2002],
 [0.55218, 0.10694, 0.22777, 0.24394, 0.13459])
```

In [ ]:

```python
# coefficient
reg.coef_
```

Out[ ]:

```
array([-0.10841385,  0.03010942,  0.03779141,  0.03916031, -0.02523263])
```

In [ ]:

```python
# r2
reg.score(X, y)
```

Out[ ]:

```
0.365064847871562
```

In [ ]:

```python
# cross validation r2 and std
get_cv(reg, X, y, loo=True)
```

Out[ ]:

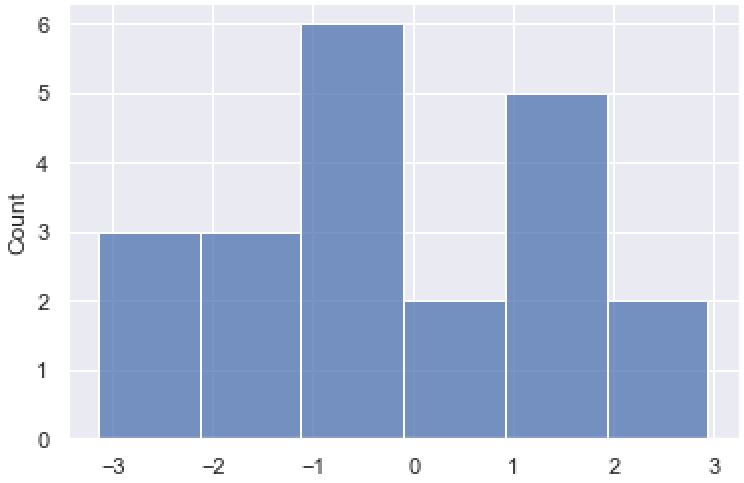
```
[0.08144,
 0.0,
 array([-1.16477859,  1.43446761, -2.28106682,  1.23100786, -0.85619942,
         1.47089549,  1.79574319,  1.87708538, -0.40099349, -0.28459951,
        -1.54635138, -1.44411554, -0.1239867 ,  0.84468749,  2.9402903 ,
         2.36256793, -3.14251197, -0.76127703,  0.03531615, -0.48762403,
        -2.40246668])]
```

In [ ]:

```
# residuals histogram
sns.histplot(get_cv(reg, X, y, loo=True)[2])
```

Out[ ]:

```
<AxesSubplot:ylabel='Count'>
```



In [ ]:

```
# global moran's I for the residuals
Moran(get_cv(reg, X, y, loo=True)[2], weight).I, Moran(get_cv(reg, X,
y, loo=True)[2], weight).p_norm
```

Out[ ]:

```
(0.035801811859351725, 0.14319436427836196)
```

In [ ]:

```
# df for annual mean per hour
hm_dep_df = dep_df.groupby(['hour', 'Site']).mean()
hm_dep_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 504 entries, (0, 'BL0') to (23, 'TH4')
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Value        504 non-null    float64
 1   month        504 non-null    float64
 2   dayofmonth   504 non-null    float64
dtypes: float64(3)
memory usage: 13.3+ KB
```

In [ ]:

```
# drop unnecessary columns
hm_dep_df.drop(['dayofmonth', 'month'], axis=1, inplace=True)

# reset index
hm_dep_df.reset_index(inplace=True)

# add explanatory variables to the df
hm_dep_df = hm_dep_df.merge(exp_df, left_on='Site', right_on='siteid')
hm_dep_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 504 entries, 0 to 503
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   hour                 504 non-null    int64  
 1   Site                 504 non-null    object 
 2   Value                504 non-null    float64
 3   siteid               504 non-null    object 
 4   UnC_area_per_len     504 non-null    float64
 5   A_area_per_len       504 non-null    float64
 6   B_area_per_len       504 non-null    float64
 7   CUn_area_per_len     504 non-null    float64
 8   Other_area_per_len   504 non-null    float64
dtypes: float64(6), int64(1), object(2)
memory usage: 39.4+ KB
```

In [ ]:

```python
# drop repetitive column
hm_dep_df.drop('Site', axis=1, inplace=True)
```

In [ ]:

```python
# check global moran's I for the 24 groups of annual means per hour
for h in range(24):
    df = hm_dep_df[hm_dep_df['hour']==h].copy()
    print("Global Moran's I for hour ", h, " is: ", Moran(df['Value'].values, weight).I)
```

```
Global Moran's I for hour  0  is:  0.06497149669113031
Global Moran's I for hour  1  is:  0.0245776965350352
Global Moran's I for hour  2  is:  0.023679068042262625
Global Moran's I for hour  3  is:  0.022407841453422502
Global Moran's I for hour  4  is:  0.010815062841263887
Global Moran's I for hour  5  is:  0.015475583335916723
Global Moran's I for hour  6  is:  0.022795098109670307
Global Moran's I for hour  7  is:  0.03812792090918269
Global Moran's I for hour  8  is:  0.061859833103432335
Global Moran's I for hour  9  is:  0.06864587291469511
Global Moran's I for hour  10  is:  0.07152289068767917
Global Moran's I for hour  11  is:  0.05695064983154796
Global Moran's I for hour  12  is:  0.044951223755330394
Global Moran's I for hour  13  is:  0.036635569274802986
Global Moran's I for hour  14  is:  0.018729390685179557
Global Moran's I for hour  15  is:  0.001011166962265878
Global Moran's I for hour  16  is:  -0.0071881322024358145
Global Moran's I for hour  17  is:  -0.008536266595833628
Global Moran's I for hour  18  is:  0.014484832240914635
Global Moran's I for hour  19  is:  0.0011745643188229072
Global Moran's I for hour  20  is:  0.01483448920752272
Global Moran's I for hour  21  is:  0.00858989331841518
Global Moran's I for hour  22  is:  0.0030212534070487417
Global Moran's I for hour  23  is:  0.012845802764135312
```

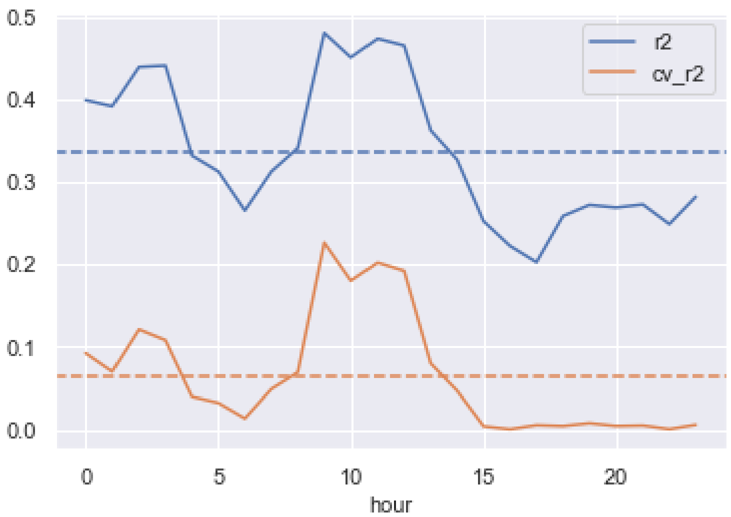
In [ ]:

```python
# linear models by each hour
hm_reg = []
```

```
for h in range(24):
    df = hm_dep_df[hm_dep_df['hour']==h].copy()

    X = df[var_names].values
    y = df['Value'].values

    mean, std = get_importance(reg, X, y, var_names)
    coef = reg.coef_.tolist() + [reg.intercept_]
    r2 = reg.score(X, y)
    cv = get_cv(reg, X, y, loo=True)

    hm_reg.append(mean+std+coef+[r2]+cv)

hm_reg = pd.DataFrame(hm_reg, columns=['fi_'+var for var in var_names]+
                      ['fi_std_'+var for var in var_names]+var_names+
                      ['intercept','r2','cv_r2','std_r2','resid'])
```

In [ ]:

```
# Fig 5 - hourly model performance
hm_reg[['r2', 'cv_r2']].plot()
plt.axhline(y=hm_reg['r2'].mean(),linestyle='--')
plt.axhline(y=hm_reg['cv_r2'].mean(), linestyle='--
',color=sns.color_palette()[1])
plt.xlabel('hour', fontsize=11)

plt.savefig('figure/Fig5.png', facecolor=None, dpi=500)

plt.show()
```



In [ ]:

```
# histogram for residuals
fig, ax = plt.subplots(4, 6, figsize=(24, 16))
```

```
for hour in range(24):
    sns.histplot(hm_reg.loc[hour,'resid'], ax=ax[hour//6, hour%6])
    ax[hour//6, hour%6].get_xaxis().set_ticks([])
    ax[hour//6, hour%6].set_title(str(hour)+'-'+str(hour+1))

plt.show()
```

In [ ]:
```
for h in range(24):
    resid = hm_reg.loc[h, 'resid']
    print("Global Moran's I for residuals for hour ", h, " is: ",
          Moran(resid, weight).I, " p-value: ", Moran(resid,
weight).p_norm)
```

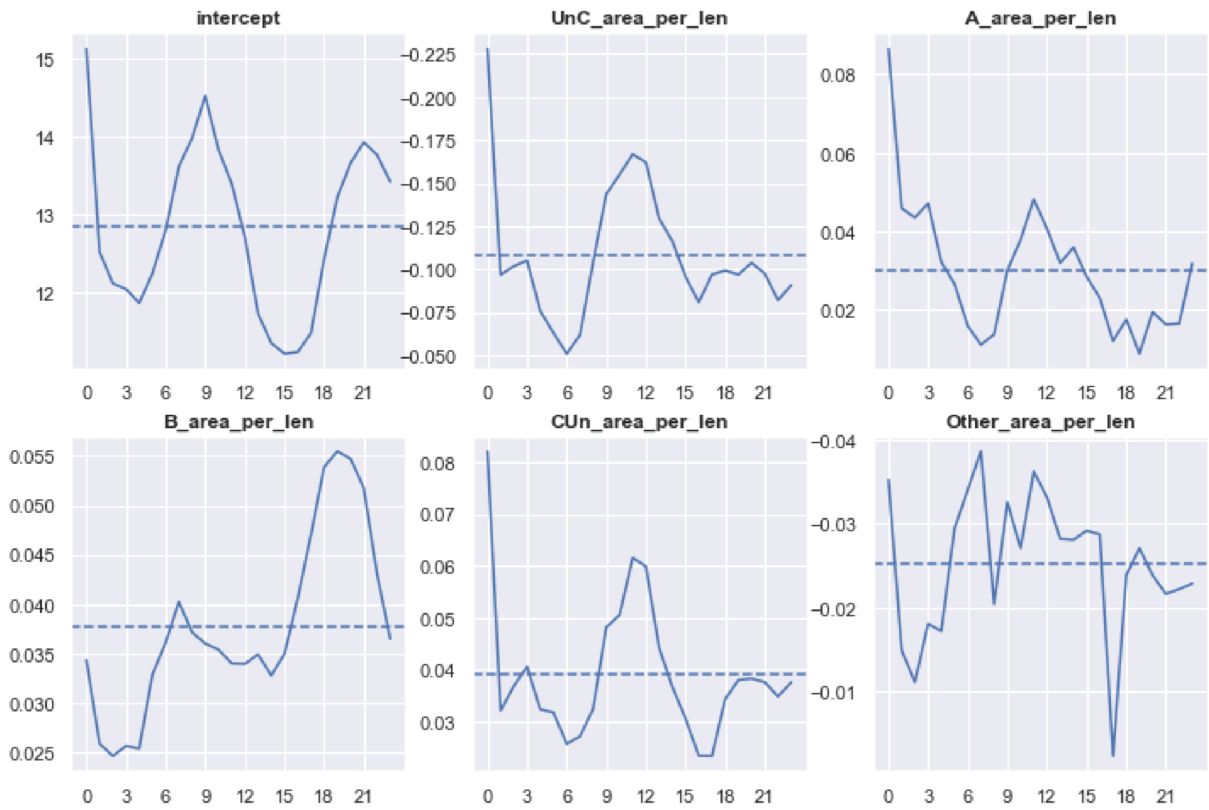
```
Global Moran's I for residuals for hour  0  is:  0.08526918977846121  p-val
ue:  0.020996993438289646
Global Moran's I for residuals for hour  1  is:  0.017680790831707732  p-va
lue:  0.24816927037495295
Global Moran's I for residuals for hour  2  is:  0.024961330542104306  p-va
lue:  0.20088574325269315
Global Moran's I for residuals for hour  3  is:  0.031908795653333044  p-va
lue:  0.16224113298007747
Global Moran's I for residuals for hour  4  is:  -0.009099531308112231  p-v
alue:  0.4852605860352657
Global Moran's I for residuals for hour  5  is:  -0.006047326898875781  p-v
alue:  0.45328778340140774
Global Moran's I for residuals for hour  6  is:  0.0039050775523988626  p-v
alue:  0.35769893977484557
Global Moran's I for residuals for hour  7  is:  0.010235774865333181  p-va
lue:  0.30405499039356254
```

```
Global Moran's I for residuals for hour  8  is:  0.035738418628806425  p-va
lue:  0.14349013902280094
Global Moran's I for residuals for hour  9  is:  0.04407216664248342  p-val
ue:  0.10846918217997881
Global Moran's I for residuals for hour  10  is:  0.05232887893251602  p-va
lue:  0.08081224902883033
Global Moran's I for residuals for hour  11  is:  0.029987947416533163  p-v
alue:  0.17231621037651967
Global Moran's I for residuals for hour  12  is:  0.007196098097735054  p-v
alue:  0.3291081746829687
Global Moran's I for residuals for hour  13  is:  -0.003195079844593924  p-
value:  0.42451559438142517
Global Moran's I for residuals for hour  14  is:  -0.020109263501667963  p-
value:  0.610042735873185
Global Moran's I for residuals for hour  15  is:  -0.031780851791542865  p-
value:  0.7559030315253836
Global Moran's I for residuals for hour  16  is:  -0.033196727462236654  p-
value:  0.7743368080970914
Global Moran's I for residuals for hour  17  is:  -0.023254069966454134  p-
value:  0.648135431764304
Global Moran's I for residuals for hour  18  is:  -0.007510291876657277  p-
value:  0.4684623664198557
Global Moran's I for residuals for hour  19  is:  -0.023575108066383316  p-
value:  0.6520787216244954
Global Moran's I for residuals for hour  20  is:  -0.006808895224931003  p-
value:  0.46115228556854815
Global Moran's I for residuals for hour  21  is:  -0.012431268660981638  p-
value:  0.5215105365580477
Global Moran's I for residuals for hour  22  is:  -0.016189530904717885  p-
value:  0.5640116558393389
Global Moran's I for residuals for hour  23  is:  -0.004729874894723929  p-
value:  0.43986323670592986
```

In [ ]:

```
# Fig 6 - hourly feature importance
fig, ax = plt.subplots(4, 6, figsize=(24, 16))

for hour in range(24):
    ax[hour//6, hour%6].bar(['fi_' + elem for elem in var_names],
                            hm_reg.loc[hour, ['fi_' + elem for elem in
var_names]].values,
                            yerr=hm_reg.loc[hour, ['fi_std_' + elem for
elem in var_names]].values)
    ax[hour//6, hour%6].get_xaxis().set_ticks([])
    ax[hour//6, hour%6].set_title(str(hour)+'-'+str(hour+1),
fontweight='bold')

plt.savefig('figure/Fig6.png', facecolor=None, dpi=500)

plt.show()
```

In [ ]:
```
# Fig 7 - coefficient viz
fig,ax=plt.subplots(2,3,figsize=(12,8))


col = ['intercept']+var_names
for i in range(len(col)):
    hm_reg[col[i]].plot(ax=ax[i//3,i%3])
    ax[i//3,i%3].set_title(col[i], fontweight='bold')
    ax[i//3,i%3].set_xticks([3*i for i in range(8)])
    ax[i//3,i%3].axhline(y=hm_reg[col[i]].mean(), linestyle='--')
    if hm_reg[col[i]].mean()<0:
        ax[i//3,i%3].invert_yaxis()


plt.savefig('figure/Fig7.png', facecolor=None, dpi=500)


plt.show()
```



In [ ]:

```python
# df for monthly mean
mm_dep_df = dep_df.groupby(['month','Site']).mean()
mm_dep_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 252 entries, (1, 'BL0') to (12, 'TH4')
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Value       252 non-null    float64
 1   hour        252 non-null    float64
 2   dayofmonth  252 non-null    float64
dtypes: float64(3)
memory usage: 6.8+ KB
```

In [ ]:

```python
# drop unnecessary columns
mm_dep_df.drop(['hour', 'dayofmonth'], axis=1, inplace=True)


# reset index
mm_dep_df.reset_index(inplace=True)


# add explanatory variables
mm_dep_df = mm_dep_df.merge(exp_df, left_on='Site', right_on='siteid')
mm_dep_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 252 entries, 0 to 251
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
```

```
 ---  ------              --------------  -----
 0    month               252 non-null    int64
 1    Site                252 non-null    object
 2    Value               252 non-null    float64
 3    siteid              252 non-null    object
 4    UnC_area_per_len    252 non-null    float64
 5    A_area_per_len      252 non-null    float64
 6    B_area_per_len      252 non-null    float64
 7    CUn_area_per_len    252 non-null    float64
 8    Other_area_per_len  252 non-null    float64
dtypes: float64(6), int64(1), object(2)
memory usage: 19.7+ KB
```

In [ ]:

```
# drop repetitive column
mm_dep_df.drop('Site', axis=1, inplace=True)
```

In [ ]:

```
# check global moran's I for the 12 groups of monthly means
for m in range(1,13):
    df = mm_dep_df[mm_dep_df['month']==m].copy()
    print("Global Moran's I for ", mlabels[m-1], "is: ",
Moran(df['Value'].values, weight).I)
```

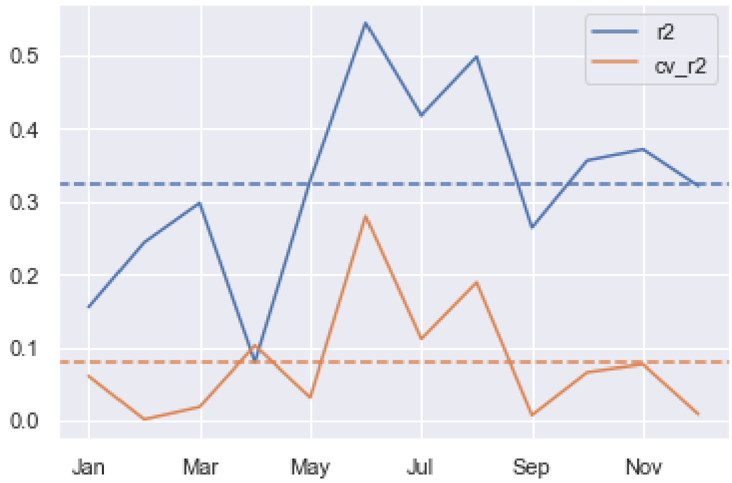
```
Global Moran's I for  Jan is:  0.003641632057737503
Global Moran's I for  Feb is:  0.0662609644851644
Global Moran's I for  Mar is:  0.13033251481287086
Global Moran's I for  Apr is:  -0.007821811529515265
Global Moran's I for  May is:  0.057814827084503126
Global Moran's I for  Jun is:  0.07317506194877035
Global Moran's I for  Jul is:  0.016624172348547708
Global Moran's I for  Aug is:  0.015710564641922425
Global Moran's I for  Sep is:  0.05979959100051212
Global Moran's I for  Oct is:  0.08902715229331153
Global Moran's I for  Nov is:  0.05306934244532968
Global Moran's I for  Dec is:  0.10564724933858408
```

In [ ]:

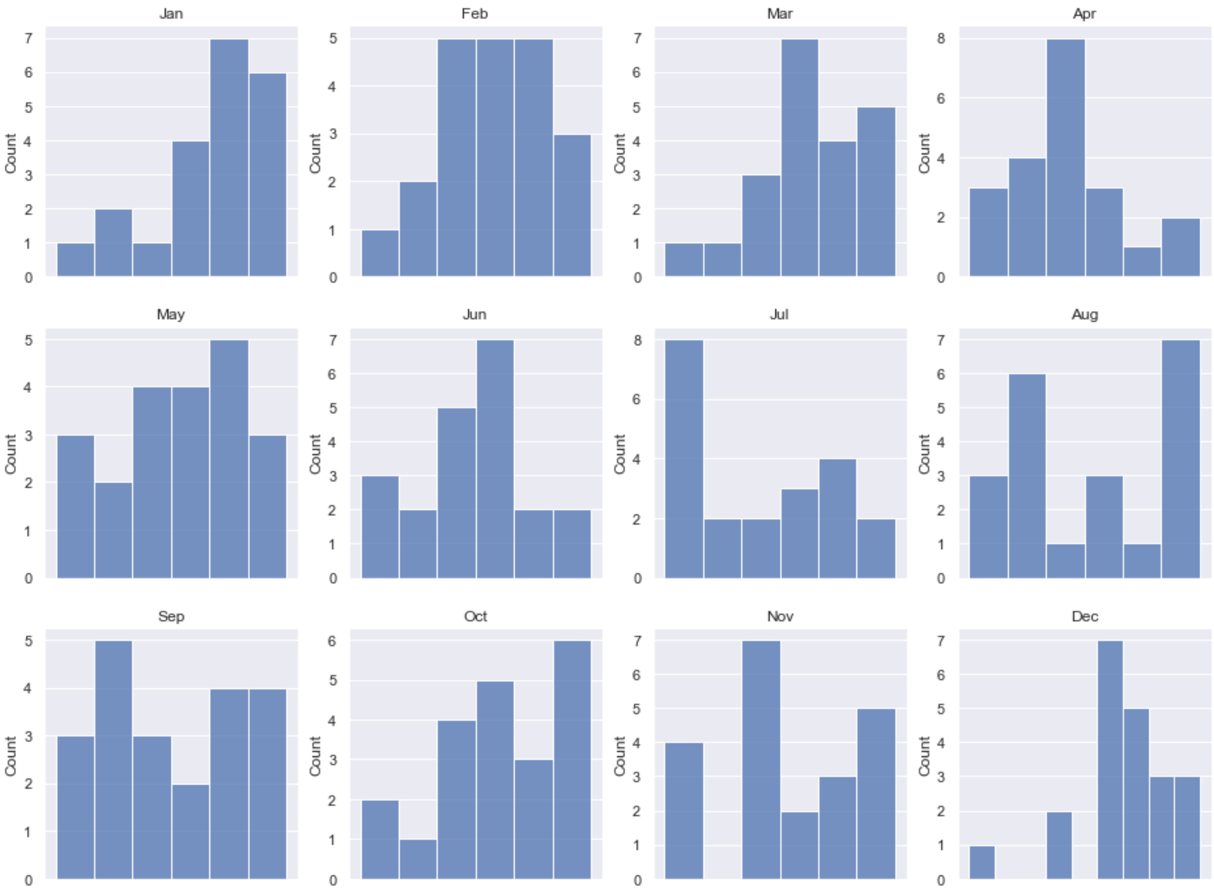
```
# linear models by each month
mm_reg = []
for m in range(1,13):
    df = mm_dep_df[mm_dep_df['month']==m].copy()

    X = df[var_names].values
    y = df['Value'].values

    mean, std = get_importance(reg, X, y, var_names)
    coef = reg.coef_.tolist() + [reg.intercept_]
    r2 = reg.score(X, y)
    cv = get_cv(reg, X, y, loo=True)

    mm_reg.append(mean+std+coef+[r2]+cv)
```

```
mm_reg = pd.DataFrame(mm_reg, columns=['fi_'+var for var in var_names]+
                      ['fi_std_'+var for var in var_names]+var_names+
                      ['intercept','r2','cv_r2','std_r2','resid'])
```

In [ ]:
```
# Fig 8 - model performance
mm_reg[['r2', 'cv_r2']].plot()

plt.axhline(y=mm_reg['r2'].mean(),linestyle='--')
plt.axhline(y=mm_reg['cv_r2'].mean(), linestyle='--
',color=sns.color_palette()[1])

plt.gca().set_xticks([0,2,4,6,8,10])
plt.gca().set_xticklabels([mlabels[2*i] for i in range(6)])

plt.savefig('figure/Fig8.png', facecolor=None, dpi=500)

plt.show()
```



In [ ]:
```
# histogram for residuals
fig, ax = plt.subplots(3, 4, figsize=(16, 12))

for month in range(12):
    sns.histplot(mm_reg.loc[month,'resid'], ax=ax[month//4, month%4])
    ax[month//4, month%4].get_xaxis().set_ticks([])
    ax[month//4, month%4].set_title(mlabels[month])

plt.show()
```



In [ ]:

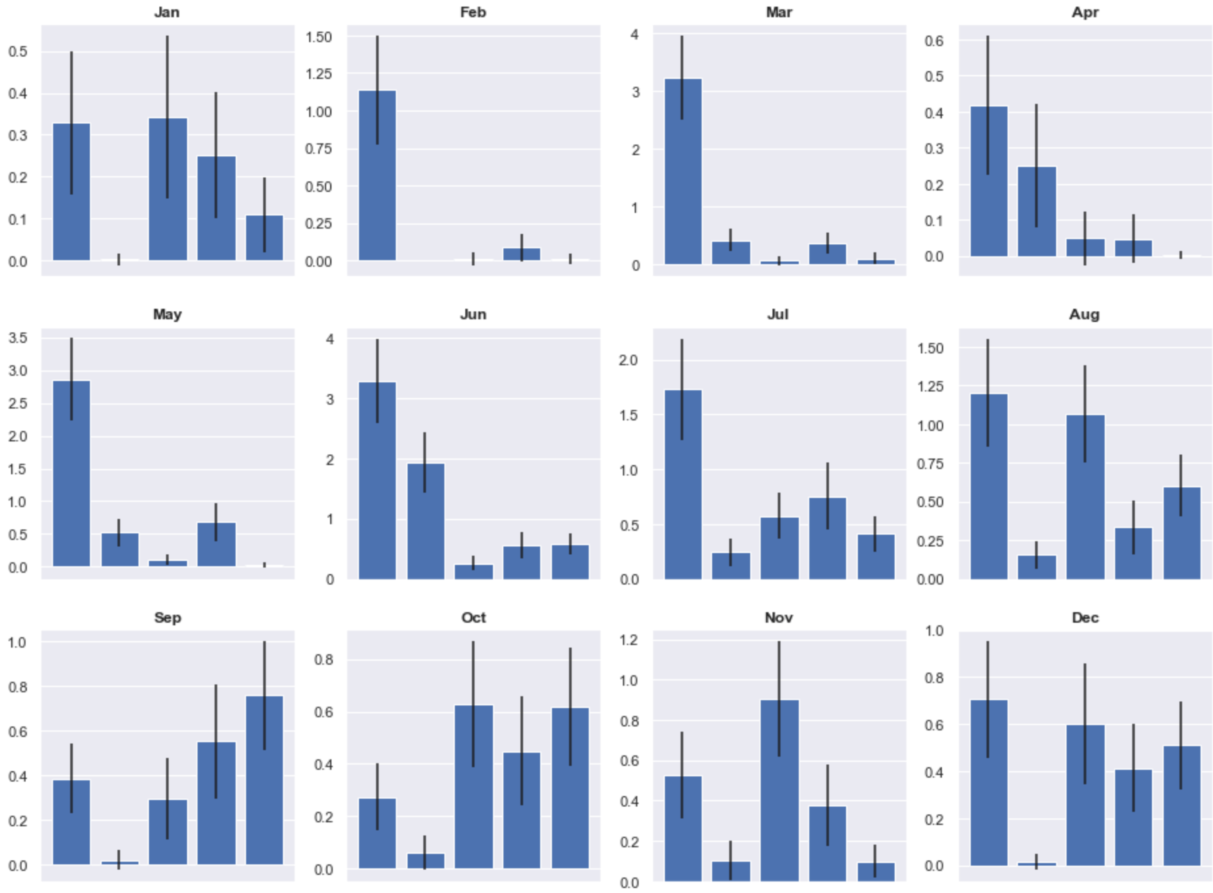
```
for m in range(12):
    resid = mm_reg.loc[m, 'resid']
    print("Global Moran's I for residuals for ", mlabels[m], " is: ",
Moran(resid, weight).I,
          " p-value: ", Moran(resid, weight).p_norm)
```

```
Global Moran's I for residuals for  Jan  is:  -0.0008782486805907636  p-val
ue:  0.4019505186489587
Global Moran's I for residuals for  Feb  is:  0.06609368032560278  p-value:
0.04760750641374334
Global Moran's I for residuals for  Mar  is:  0.0827955347494292  p-value:
0.023461692908614662
Global Moran's I for residuals for  Apr  is:  0.014224891199051569  p-valu
e:  0.27314836431004474
Global Moran's I for residuals for  May  is:  0.014277374642435135  p-valu
e:  0.27275659685651577
Global Moran's I for residuals for  Jun  is:  0.08420680063133579  p-value:
0.022026450070490977
Global Moran's I for residuals for  Jul  is:  0.022846624416685758  p-valu
e:  0.21388632256854834
Global Moran's I for residuals for  Aug  is:  0.0185852020202588  p-value:
0.24190477950327383
Global Moran's I for residuals for  Sep  is:  0.017548951329757163  p-valu
e:  0.24909186377633974
Global Moran's I for residuals for  Oct  is:  0.029116238322654784  p-valu
e:  0.17703988607960386
Global Moran's I for residuals for  Nov  is:  0.02780130619328  p-value:
0.18434685142863128
```

```
Global Moran's I for residuals for  Dec  is:  0.11965867076691691  p-value: 0.0037938678648252733
```

In [ ]:

```python
# Fig 9 - plot feature importance
fig, ax = plt.subplots(3, 4, figsize=(16, 12))

for month in range(12):
    ax[month//4, month%4].bar(['fi_' + elem for elem in var_names],
                              mm_reg.loc[month, ['fi_' + elem for elem in var_names]].values,
                              yerr=mm_reg.loc[month, ['fi_std_' + elem for elem in var_names]].values)
    ax[month//4, month%4].get_xaxis().set_ticks([])
    ax[month//4, month%4].set_title(mlabels[month], fontweight='bold')

plt.savefig('figure/Fig9.png', facecolor=None, dpi=500)

plt.show()
```



In [ ]:

```python
# Fig 10 - coefficient viz
fig,ax=plt.subplots(2,3,figsize=(12,8))

for i in range(len(col)):
    mm_reg[col[i]].plot(ax=ax[i//3,i%3])
```

```
    ax[i//3,i%3].set_title(col[i], fontweight='bold')
    ax[i//3,i%3].set_xticks([0,2,4,6,8,10])
    ax[i//3,i%3].set_xticklabels([mlabels[2*i] for i in range(6)])
    ax[i//3,i%3].axhline(y=mm_reg[col[i]].mean(), linestyle='--')
    if mm_reg[col[i]].max()<0:
        ax[i//3,i%3].invert_yaxis()

plt.savefig('figure/Fig10.png', facecolor=None, dpi=500)

plt.show()
```