Technical note: saqgetr R package (Version 0.1)

Stuart K. Grange [1,2,*]

October 14, 2019

[1] Empa, Swiss Federal Laboratories for Materials Science and Technology, Überlandstrasse 129, 8600 Dübendorf, Switzerland

[2] Wolfson Atmospheric Chemistry Laboratories, University of York, York, YO10 5DD, United Kingdom

[*] stuart.grange@empa.ch

1 Introduction

The **saqgetr** R package acts as an interface to access air quality monitoring data and is available on the Comprehensive R Archive Network (CRAN).[1] The vast majority of the accessible monitoring data has been collected in Europe for compliance purposes and are available openly from the European Commission's Airbase and air quality e-reporting (AQER) repositories.[2,3] Although these European Commission repositories are publicly accessible, to use the data efficiently they require processing into a harmonised form with consistent and careful treatment of the observations and metadata. This re-databasing has been conducted by myself and the database which has resulted in what is known as **smonitor** Europe[4] and has seen much use in my research activities.[5–10] The direct sharing of the **smonitor** Europe database would be challenging and therefore, another layer of infrastructure has been created with a web server provided by Ricardo Energy & Environment[*] hosting compressed, plain text files which can be accessed by anyone. The **saqgetr** R package contains a collection of importing functions to make the use of these data simple and fast for R users.

1.1 Objectives

The primary objective of this note is to document the data sources which have been used to form the database so it is clear where all data have been sourced from for users of **saqgetr**. A secondary objective is to outline the basic usage of **saqgetr**. For those who are interested in the underlying **smonitor** data model and how it is implemented, another technical note[11] and my PhD thesis (chapter 3)[9] are the best resources for this.

2 Air quality data sources

The data accessible by **saqgetr** can be generally found in other, publicly available data repositories. What makes **saqgetr** useful is that the multiple data sources have been combined in

[*] `https://ee.ricardo.com/air-quality`

a single place with a common data model. This allows the importing of large numbers of observations with a few lines of code. Furthermore, the consistency makes the use and analysis of data from different sources uniform and much easier. The vast majority of the data accessible with **saqgetr** are found in the European Environment Agency's (EEA) repositories which are maintained for compliance to the air quality reporting directives 2004/107/EC and 2008/50/EC.[2,3] The first EEA repository is called AirBase which which contains data between 1969 and 2012 inclusive.[14,15] From 2013 onwards, the AirBase reporting system was superseded by what is known as Air Quality e-Reporting (AQER).[16] AQER forms a small piece of the larger EEA central data repository (CDR) which contains data and information for a very wide range of topics for the European Union (EU) and its member, or associated states.

2.1 Airbase

AirBase Version 8 data were downloaded in bulk from the EEA data portal and the observations and aggregations are available per country/state as `.zip` archives.[14] † The files containing observational data are supplied in simple, tabular text files. Two helper tables are required to make complete sense of the observations which are called the station and measurement configurations files. These two files are approximately equivalent to the **smonitor** `sites` and `processes` tables. Because AirBase has been replaced with AQER, it is likely that these data will remain static into the future unless a large systematic error is discovered and needs to be addressed. There are a few known outstanding issues with the AirBase data. The two most notable is that for a few member states, the 2012 year is completely absent from the repository, for example Germany and France, and the time zone used for each member state is undocumented.

2.2 Air quality e-Reporting

After 2012, EU member states were required to submit AQER data to the CDR in specified formats to comply with the air quality reporting directives. Unfortunately, the file format used for this process is XML. This means that both the creation, and reading of these files is not a trivial task. The complication of the XML documents' schemas has created many issues where many member states requite several attempts to generate the correct XML documents. The diversity of submissions has made it difficult and time consuming to develop generic functions which read all AQER data files without errors.

† `https://www.eea.europa.eu/data-and-maps/data/airbase-the-european-air-quality-database-8#tab-data-by-country`

Within the AQER system, there are a number of observational units termed "data flows". There are many data flows, but the most relevant ones for air quality observational data are:

Data flow D Information on the assessment methods (site and process metadata).

Data flow E1a Information on primary validated assessment data measurements (validated observations).

Data flow E2a Information on primary up-to-date assessment data measurements (near-real time observations).

An example of a complete set of AQER data flows can be found for the United Kingdom AQER page[‡], but most member states have a similar collection.

2.3 The E1a and E2a download portal

Originally, XML files were downloaded which were of interest from the CDR and the files were scraped make a complete data model for **smonitor** Europe.[9] However, in late-2018, the EEA released a download portal which mostly removes the need to scrape these XML documents directly.[17][§] Having access to this portal is a huge advantage when compared to the previous situation. The download portal supplies `.csv` files to a user which have been sourced from the member state XML submissions and created by the EEA. To update **smonitor** Europe, this portal is interacted with programatically with another R package called **aqerimportr**.[18] The access to this download portal also enables access to the E2a data flow (near-real time observations) which is not possible by using the CDR repository alone and is probably why the download portal exists.

A filtered version of the site and process metadata (Data Flow D) is also available from the download portal. This filtered metadata file is missing some pieces of very useful information such as site names. Therefore, the download and scraping of some XML documents to satisfy the **smonitor** data model is still done, but this process is no longer a critical component to maintain **smonitor** Europe.

E1a data are validated and are considered the final, and correct air quality monitoring record while E2a observations are supplied from automated systems and are always provisional or indicative. E2a data are deleted and completely superseded when the E1a data are delivered and checked by the EEA. E1a data are submitted annually and the deadline for submission by the member states is September 30. This means that there is a ten-month delay after a year has

[‡]http://cdr.eionet.europa.eu/gb/eu/aqd

[§]http://discomap.eea.europa.eu/map/fme/AirQualityExport.htm ended before validated observations become available. There are additional delays while the EEA check the submissions before they are copied to the download portal too.

These details on the E1a and E2a data flow availability are important to know when using data from **saqgetr**. I endeavour to update **smonitor** Europe as soon as new E1a deliveries available and to keep E2a observations up-to-date within about a month. It will be possible to improve the punctuality of the E2a updates in the future, but the download portal, to-date, has been rather variable on what the latest observations it can supply (it can be up to six weeks behind). So for now, scheduling daily updating and copying tasks would not be particularly beneficial.

2.4 Other data sources

The other data sources in use and are accessible by **saqgetr** are: the Centre for Environmental Data Analysis (CEDA),[19] GAW World Data Centre for Reactive Gases (hosted by EBAS),[20] and the World Data Centre for Greenhouse Gases (WDCGG) is a World Data Centre (WDC) (WDCGG).[21] These data sources are only minor contributions to **saqgetr** but may grow in their importance in the future as my research progresses.

2.5 Known source data issues

AirBase time zones The time zones used for the member states in the AirBase repository are undocumented. UTC has been used but this is likely to be incorrect for some member states' data and is rather difficult to test.

AirBase 2012 observations Some member states observational data for 2012 are absent from the AirBase repository, for example Germany and France.

NO_x, NO_2, and NO reporting Many member states no longer report the full complement of NO_x, NO_2, and NO. This is frustrating to air quality scientists because at least two of these need to be present to conduct many analyses.

Polish site names The special characters in the Polish site names have not been encoded correctly and therefore are incorrectly represented.

2.6 To-do

Add spatial data to saqgetr European air quality management zones are available for most pollutants in AQER's Data Flow B. These spatial polygons should be accessible by **saqgetr**.

3 Basic saqgetr usage

3.1 Installation

The **saqgetr** package can be installed in the standard way from CRAN in an R session with `install.packages("saqgetr")`. The development of **saqgetr** is hosted on GitHub so the development version be installed with the help of the **remotes** package: `remotes::install_github("skgrange/saqgetr")`. **saqgetr** has no unusual dependencies and does not include any code which requires compilation so should not raise obscure installation issues for modern systems.

3.2 Work flow and function use

The workflow when using **saqgetr** will generally take these steps:

1. Determine what sites' observations are desired and identify what these sites are called.

2. Determine what variables/pollutants are available for the sites of interest for a particular time period, as well as what type of observations they are. For example, are hourly observations available for PM_{10} or are only daily observations accessible?

3. Get or import the observations of interest.

4. Clean or prepare data for your data analysis requirements. This will usually involve removing invalid data and often reshaping data.

5. Continue to do some interesting data analysis. This is the primary goal of **saqgetr** to help data users for this purpose.

The steps above are done by functions which names should make it clear what they do: `get_saq_sites`, `get_saq_processes`, `get_saq_observations`, and `saq_clean_observations`.

3.2.1 Getting site information

Before observations can be imported with **saqgetr**, site codes must be identified. Getting information about what air quality monitoring site/stations/facilities are serviced is done with the use of `get_saq_sites` (Listing 1). The tibble (data frame/table) which is returned by `get_saq_sites` has many variables which are generally useful to know including coordinates, site names, site type classification, different site identifiers, dates of when observations begin and end in the database, and where the observations have been sourced from. The unique site identifier used by **saqgetr** is the the `site` variable (the first column). In the **smonitor** framework, a `site` string always starts with the country's ISO 3166-1 alpha-2 code (but is lower-case), followed with an integer, and finally a trailing letter which usually indicates the original site classification but this is not a robust indicator. The site code remains identical among the many tables which are imported by **saqgetr** so using the site metadata alongside the observations is easily achieved with joins. For those who prefer to use a map to find sites, an interactive, but rather slow map is hosted online.[¶]

Listing 1: An example of using **saqgetr**'s `get_saq_sites` function.

```r
# Load packages which are required for this note
library(dplyr)
library(saqgetr)

# Import site information
data_sites <- get_saq_sites()

# Glimpse tibble
glimpse(data_sites)

#> Observations: 9,026
#> Variables: 16
#> $ site              <chr> "ad0942a", "ad0944a", "ad0945a", "al0201a", "a
#> $ site_name         <chr> "Fixa", "Fixa oz", "Estacional oz Envalira", "
#> $ latitude          <dbl> 42.50969, 42.51694, 42.53488, 41.33027, 41.345
#> $ longitude         <dbl> 1.539138, 1.565250, 1.716986, 19.821772, 19.85
#> $ elevation         <dbl> 1080, 1637, 2515, 162, 207, 848, 25, 1, 13, 15
#> $ country           <chr> "andorra", "andorra", "andorra", "albania", "a
#> $ country_iso_code  <chr> "AD", "AD", "AD", "AL", "AL", "AL", "AL", "AL"
#> $ site_type         <chr> "background", "background", "background", "tra
#> $ site_area         <chr> "urban", "rural", "rural", "urban", "suburban"
#> $ date_start        <dttm> 2017-12-31 23:00:00, 2017-12-31 23:00:00, 201
#> $ date_end          <dttm> 2019-09-21 10:00:00, 2019-09-21 10:00:00, 201
#> $ network           <chr> "NET-AD001A", "NET-AD001A", "NET-AD001A", "NET
#> $ eu_code           <chr> "STA-AD0942A", "STA-AD0944A", "STA-AD0945A", "
#> $ eoi_code          <chr> "AD0942A", "AD0944A", "AD0945A", "STA-AL0201A"
#> $ observation_count <dbl> 97864, 13879, 3450, 145816, 119022, 0, 0, 0, 0
#> $ data_source       <chr> "aqer:e1a; aqer:e2a", "aqer:e1a; aqer:e2a", "a
```

[¶]http://skgrange.github.io/www/maps/smonitor_europe_sites/smonitor_europe_sites.html

3.2.2 Getting process information

A **smonitor** process is an integer key which represents a unique time series and is used by by **saqgetr**.[4,9] A unique time series can be thought of a site-pollutant combination, but most processes will usually be more specific. For example, in the **smonitor** Europe database, there are separate processes for different averaging or aggregation periods and different data sources. Therefore, for a single site and pollutant, there will often be several processes which represent different aggregation periods, instrumentation changes, and differing data sources. These data are imported by using `get_saq_processes` (Listing 2). Data on the processes is often not not necessary to use, but there are some very useful variables in this table which are of interest to most data users.

Listing 2: An example of using **saqgetr**'s `get_saq_processes` function.

```r
# Get processes
data_processes <- get_saq_processes()

# Glimpse tibble
glimpse(data_processes)

#> Observations: 172,841
#> Variables: 15
#> $ process           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
#> $ site              <chr> "al0201a", "al0201a", "al0201a", "al0201a", "a
#> $ variable          <chr> "so2", "so2", "pm10", "pm10", "o3", "o3", "o3"
#> $ variable_long     <chr> "Sulphur dioxide (air)", "Sulphur dioxide (air
#> $ period            <chr> "day", "hour", "day", "hour", "day", "dymax",
#> $ unit              <chr> "ug.m-3", "ug.m-3", "ug.m-3", "ug.m-3", "ug.m-
#> $ date_start        <dttm> NA, 2011-01-01 00:00:00, 2011-01-01 00:00:00,
#> $ date_end          <dttm> NA, 2011-12-31 23:00:00, 2012-12-30 00:00:00,
#> $ sample            <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
#> $ sampling_point    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
#> $ sampling_process  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
#> $ observed_property <int> 1, 1, 5, 5, 7, 7, 7, 7, 8, 8, 9, 9, 10, 10, 10
#> $ group_code        <int> 100, 100, 100, 100, 100, 100, 100, 100, 100, 1
#> $ data_source       <chr> "airbase", "airbase", "airbase", "airbase", "a
#> $ observation_count <dbl> 0, 6806, 729, 17336, 352, 352, 16413, 8358, 69
```

3.2.3 Getting observations

Once a user has identified one or more sites which observations are desired for (see Listing 1), the `get_saq_observations` function is used to import these data. The examples below use the gb0919a site which is York Fishergate, an urban-traffic site located in the City of York in the North of England (Listing 3). The tibble which is returned by get_saq_observations contains most things which are needed to inform a data user about the observations such as start and end dates (end dates are not available in the earlier European repositories) and the units in use. All variables will be imported with correct data types.

The summary integer in the observational data represents what aggregation and time period the observation is formed from. The most common summaries are 1 and 20 which represent hourly and daily means respectively. Validity integers represent observation validity where positive integers mean the observation is valid while values of zero and below indicate invalid observations. For a full list of these integer codes, see Listing 6.

Listing 3: An example of using **saqgetr**'s get_saq_observations function to get many years of data for the gb0919a site (York Fishergate).

```
# Get air quality monitoring data for a York site
data_york <- get_saq_observations(site = "gb0919a", start = 2005)

# Glimpse tibble
glimpse(data_york)

#> Observations: 377,296
#> Variables: 9
#> $ date     <dttm> 2008-01-01, 2008-01-02, 2008-01-03, 2008-01-04, 2008-0
#> $ date_end <dttm> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
#> $ site     <chr> "gb0919a", "gb0919a", "gb0919a", "gb0919a", "gb0919a",
#> $ variable <chr> "pm10", "pm10", "pm10", "pm10", "pm10", "pm10", "pm10",
#> $ process  <int> 62392, 62392, 62392, 62392, 62392, 62392, 62392, 62392,
#> $ summary  <int> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20,
#> $ validity <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1
#> $ unit     <chr> "ug.m-3", "ug.m-3", "ug.m-3", "ug.m-3", "ug.m-3", "ug.m
#> $ value    <dbl> 21.625, 22.708, 24.667, 21.833, 24.000, 29.875, 16.833,
```

Listing 3 uses the start argument to get observations from many years in the past. By default, get_saq_observations will only return the current year's data. An end argument is also available to be used. The default time zone used for importing is set as UTC, but there are two helper functions, tz_central and tz_eastern which contain the time zone string used in Central and Eastern Europe (UTC+1 and UTC+2 respectively with no daylight saving clock changes) which will be more appropriate for most places in Europe.

To import many sites at the same time, simply give get_saq_observations a vector of site names (Listing 4). Have an appreciation that **saqgetr** accesses a large number of observations however, and if you are to pass tens or hundreds of sites to `get_saq_observations`, you may exhaust your system's physical memory.

Listing 4: An example of using **saqgetr**'s `get_saq_observations` function to get 10 million observations from two long running sites in the United Kingdom.

```
# Get 10 million observations, verbose is used to give an indication on
# what is occuring
data_large_ish <- get_saq_observations(
  site = c("gb0036r", "gb0682a"),
  start = 1960,
  verbose = TRUE
)

# Glimpse tibble
glimpse(data_large_ish)

#> Observations: 10,009,837
#> Variables: 9
#> $ date     <dttm> 1976-06-22, 1976-06-23, 1976-06-24, 1976-06-25, 1976-0
#> $ date_end <dttm> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
#> $ site     <chr> "gb0036r", "gb0036r", "gb0036r", "gb0036r", "gb0036r",
#> $ variable <chr> "o3", "o3", "o3", "o3", "o3", "o3", "o3", "o3", "o3", "
#> $ process  <int> 57306, 57306, 57306, 57306, 57306, 57306, 57306, 57306,
#> $ summary  <int> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20,
#> $ validity <int> 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
#> $ unit     <chr> "ug.m-3", "ug.m-3", "ug.m-3", "ug.m-3", "ug.m-3", "ug.m
#> $ value    <dbl> 131.667, 154.917, 112.833, 166.833, 256.286, 135.583, 2
```

3.2.4 Preparing observations

In most cases, a user will only want to use valid observations for analysis. Invalid observations can be removed by filtering after the importing stage, or by setting the `valid_only` argument to `TRUE` in the `get_saq_observations` function.

For some analyses, especially those done in the **openair** package,[12,13] the observational data will be more conveniently interacted with when the pollutants form their own columns. Operations such as this are called reshaping and is a rather large and tricky topic in itself, but the `saq_clean_observations` function will do this reshaping for you with the `spread` argument and will make the table "wider" when set to `TRUE`. Using observations in this so-called wide format only really makes sense when you are analysing pollutants or variables at a uniform and fixed time resolution such as an hourly or daily time series. Therefore, saq_clean_observations will filter the imported observations to such a time period before reshaping. This wide format is also unable to store duplicate date-variable pairs and will remove data if required, but will raise a warning if this is necessary.

Listing 5: An example of using **saqgetr**'s saq_clean_observations function to produce a time series which contains only hourly valid data and is stored in a "wider" format.

```
# Get only valid hourly data and reshape (spread)
data_york_spread <- data_york %>%
  saq_clean_observations(summary = "hour", valid_only = TRUE, spread = TRUE)
#> Warning: Duplicated date-site-variable combinations detected, observations
#> have been removed...

# Glimpse tibble, columns are now variables
glimpse(data_york_spread)

#> Observations: 100,062
#> Variables: 8
#> $ date     <dttm> 2008-01-01 00:00:00, 2008-01-01 01:00:00, 2008-01-01 0
#> $ date_end <dttm> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
#> $ site     <chr> "gb0919a", "gb0919a", "gb0919a", "gb0919a", "gb0919a",
#> $ no       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
#> $ no2      <dbl> 36, 32, 31, 31, 29, 31, 32, 32, 27, 25, 25, 25, 29, 36,
#> $ nox      <dbl> 50, 40, 40, 38, 34, 36, 40, 42, 40, 34, 40, 38, 52, 74,
#> $ pm10     <dbl> 23, 23, 23, 22, 20, 21, 22, 26, 26, 18, 17, 16, 23, 20,
#> $ pm2.5    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
```

3.3 Helper tables

The integers codes used for the summary and validity variables can be decoded by using the two helper functions: get_saq_summaries and get_saq_validity (Listing 6).

Listing 6: Examples of using **saqgetr**'s `get_saq_summaries` and `get_saq_validity` functions.

```
# Get other helper tables
# Summary integers
data_summary_integers <- get_saq_summaries() %>%
  print(n = Inf)
#> # A tibble: 20 x 2
#>    averaging_period summary
#>    <chr>              <int>
#>  1 hour                   1
#>  2 day                   20
#>  3 week                  90
#>  4 var                   91
#>  5 month                 92
#>  6 fortnight             93
#>  7 3month                94
#>  8 2month                95
#>  9 2day                  96
#> 10 3day                  97
#> 11 2week                 98
#> 12 4week                 99
#> 13 3hour                100
#> 14 8hour                101
#> 15 hour8                101
#> 16 year                 102
#> 17 dymax                 21
#> 18 quarter              103
#> 19 other                 91
#> 20 n-hour               104

# Validity integers
data_validity_integers <- get_saq_validity() %>%
  print(n = Inf)
#> # A tibble: 6 x 4
#>   validity valid description                                 notes
#>      <int> <lgl> <chr>                                       <chr>
#> 1       NA FALSE data is considered to be invalid due to the from aqer
#> 2       -1 FALSE invalid due to other circumstances or data  from aqer
#> 3        0 FALSE invalid
#> 4        1 TRUE  <NA>                                        from aqer
#> 5        2 TRUE  valid but below detection limit measurement from aqer
#> 6        3 TRUE  valid but below detection limit and number  from aqer
```

3.4 Useful summaries

For first analyses, quick checks, or analyses involving hundreds or thousands of sites, it is often useful to look at monthly or annual averages before the higher resolution time series. To accommodate this, simple means have been calculated for all hourly and daily processes for each site in the **smonitor** Europe database. The granularity of the individual processes is lost by this operation and these summaries are not provided for compliance purposes. They are available to enable rapid access to common aggregations for data users. To import these aggregations, use the `get_saq_simple_summaries` function with either the summary argument set to `"annual_mean"` or `"monthly_mean"` (Listing 7).

Listing 7: Examples of using **saqgetr**'s `get_saq_simple_summaries` to import annual and monthly means.

```
# Get annual means
data_annual <- get_saq_simple_summaries(summary = "annual_mean")

# Glimpse tibble
glimpse(data_annual)
#> Observations: 655,234
#> Variables: 8
#> $ date           <dttm> 2013-01-01, 2014-01-01, 2015-01-01, 2016-01-01,
#> $ date_end       <dttm> 2013-12-31 23:59:59, 2014-12-31 23:59:59, 2015-1
#> $ site           <chr> "ad0942a", "ad0942a", "ad0942a", "ad0942a", "ad09
#> $ variable       <chr> "co", "co", "co", "co", "co", "co", "co", "no", "
#> $ summary_source <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
#> $ summary        <int> 102, 102, 102, 102, 102, 102, 102, 102, 102, 102,
#> $ count          <dbl> 1, 8438, 8385, 8171, 8441, 8217, 5990, 1, 8310, 8
#> $ value          <dbl> 0.5000000, 0.3224579, 0.3582230, 0.3168768, 0.259

# What was York Fishergate's (hourly) PM10 concentraion in 2017?
data_annual %>%
  filter(site == "gb0682a",
         lubridate::year(date) == 2017L,
         variable == "pm10",
         summary_source == 1L) %>%
  select(date,
         site,
         variable,
         count,
         value)

#> # A tibble: 1 x 5
#>   date                site    variable count value
#>   <dttm>              <chr>   <chr>    <dbl> <dbl>
#> 1 2017-01-01 00:00:00 gb0682a pm10      8442  23.8

# Get monthly means too, quite a bit of data to download so may be slow
data_monthly <- get_saq_simple_summaries(summary = "monthly_mean")
```

Acknowledgements

I thank Sean Christiansen, Paul Willis, and Trevor Davis from Ricardo Energy & Environment's Air Quality & Environment Practice for their assistance and maintenance of the web server where monitoring data is stored and accessed. I am financially supported by the the Swiss Federal Office for the Environment (FOEN) and am also supported by the Natural Environment Research Council (NERC) while holding associate status at the University of York.

4 References

[1] Grange, Stuart K. ***saqgetr**: Import Air Quality Monitoring Data in a Fast and Easy Way*. R package. 2019. URL: https://cran.r-project.org/web/packages/saqgetr/index.html.

[2] European Parliament. Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe. *Official Journal of the European Union* (2008). URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32008L0050.

[3] European Parliament. 2011/850/EU: Commission Implementing Decision of 12 December 2011 laying down rules for Directives 2004/107/EC and 2008/50/EC of the European Parliament and of the Council as regards the reciprocal exchange of information and reporting on ambient air quality (notified under document C(2011) 9068). *Official Journal of the European Union* (2011). Document 32011D0850. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32011D0850.

[4] Grange, Stuart K. ***smonitor**: A framework and a collection of functions to allow for maintenance of air quality monitoring data*. 2018. URL: https://github.com/skgrange/smonitor.

[5] Grange, Stuart K., Lewis, Alastair C., Moller, Sarah J., and Carslaw, David C. Lower vehicular primary emissions of NO_2 in Europe than assumed in policy projections. *Nature Geoscience* 10.12 (2017), pp. 914–918. URL: https://doi.org/10.1038/s41561-017-0009-0.

[6] Grange, S. K., Carslaw, D. C., Lewis, A. C., Boleti, E., and Hueglin, C. Random forest meteorological normalisation models for Swiss PM_{10} trend analysis. *Atmospheric Chemistry and Physics* 18.9 (2018), pp. 6223–6239. DOI: https://doi.org/10.5194/acp-18-6223-2018. URL: https://www.atmos-chem-phys.net/18/6223/2018/.

[7] Hu, L., Keller, C. A., Long, M. S., Sherwen, T., Auer, B., Da Silva, A., Nielsen, J. E., Pawson, S., Thompson, M. A., Trayanov, A. L., Travis, K. R., Grange, S. K., Evans, M. J., and Jacob, D. J. Global simulation of tropospheric chemistry at 12.5 km resolution: performance and evaluation of the GEOS-Chem chemical module (v10-1) within the NASA GEOS Earth system model (GEOS-5 ESM). *Geoscientific Model Development* 11.11 (2018), pp. 4603–4620. URL: https://www.geosci-model-dev.net/11/4603/2018/.

[8] Grange, Stuart K. and Carslaw, David C. Using meteorological normalisation to detect interventions in air quality time series. *Science of the Total Environment* 653 (2019), pp. 578–588. URL: http://www.sciencedirect.com/science/article/pii/S004896971834244X.

[9] Grange, Stuart K. Development of Data Analytic Approaches for Air Quality Data. PhD thesis. Chemistry, The University of York, 2019. URL: http://etheses.whiterose.ac.uk/23306.

[10] Grange, Stuart K., Farren, Naomi J., Vaughan, Adam R., Rose, Rebecca A., and Carslaw, David C. Strong Temperature Dependence for Light-Duty Diesel Vehicle NO_x Emissions. *Environmental Science & Technology* 53.11 (2019), pp. 6587–6596. DOI: 10.1021/acs.est.9b01024. URL: https://doi.org/10.1021/acs.est.9b01024.

[11] Grange, Stuart K. *Technical note: **smonitor** Europe*. Tech. rep. Wolfson Atmospheric Chemistry Laboratories, University of York, 2017. DOI: https://doi.org/10.13140/RG.2.2.20555.49448/1. URL: https://doi.org/10.13140/RG.2.2.20555.49448/1.

[12] Carslaw, David C. and Ropkins, Karl. *openair* — An R package for air quality data analysis. *Environmental Modelling & Software* 27–28.0 (2012), pp. 52–61. URL: http://www.sciencedirect.com/science/article/pii/S1364815211002064.

[13] Carslaw, David and Ropkins, Karl. *openair: Open-source tools for the analysis of air pollution data*. 2015.

[14] European Environment Agency. *AirBase – The European air quality database (Version 8)*. 2014. URL: https://www.eea.europa.eu/data-and-maps/data/airbase-the-european-air-quality-database-8.

[15] Simoens, David. AirBase version 8 data products on EEA data service. European Environment Agency. 2014. URL: http://www.eea.europa.eu/data-and-maps/data/airbase-the-european-air-quality-database-8.

[16] European Environment Agency. *Eionet Central Data Repository*. 2017. URL: http://cdr.eionet.europa.eu/.

[17] European Environment Agency. Download of air quality data. Download service for E1a and E2a data. 2019. URL: http://discomap.eea.europa.eu/map/fme/AirQualityExport.htm.

[18] Grange, Stuart K. ***aqerimportr**: Import European Air Quality e-Reporting (AQER) Data*. R package. 2019. URL: https://github.com/skgrange/aqerimportr.

[19] Centre for Environmental Data Analysis. Centre for Environmental Data Analysis (CEDA). Data and information services for environmental science. 2017. URL: http://www.ceda.ac.uk/.

[20] Norwegian Institute for Air Research. EBAS. 2017. URL: http://ebas.nilu.no/.

[21] Japan Meteorological Agency. World Data Centre for Greenhouse Gases (WDCGG). 2018. URL: https://gaw.kishou.go.jp.