Extended vignette for **EEAaq: Handle Air Quality Data from the European Environment Agency Data Portal**

Paolo Maranzano, University of Milano-Bicocca, Italy, paolo.maranzano@unimib.it
Riccardo Borgoni, University of Milano-Bicocca, Italy, riccardo.borgoni@unimib.it
Agostino Tassan Mazzocco, University of Milano-Bicocca, Italy
Samir Doghmi, University of Milano-Bicocca, Italy

05 febbraio 2025

The **EEAaq package** allows users to retrieve air quality data for multiple geographical zones, pollutants, and time periods in a single request. Queries are submitted as lists, which enables flexibility in specifying combinations of parameters.

```
library(EEAaq)
library(tidyverse)
```

```
## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

EEAaq_get_data

Below we demonstrate the use of query by using different combinations of user-defined arguments.

Retrieve NO_2 data for a specific municipality (LAU zone) given its unique identifier `LAU_ID`

```
data_lau <- EEAaq::EEAaq_get_data(
  zone_name = "15146",       # LAU zone code
  NUTS_level = "LAU",        # NUTS level
  LAU_ISO = "IT",            # Country code for Italy
  pollutants = "PM10",       # Pollutant
  from = "2022-01-01",       # Start date
  to = "2023-12-31",         # End date
  verbose = FALSE            # Print detailed progress
)
```

```
# Preview the first few rows of the dataset
head(data_lau)
```

```
## # A tibble: 6 x 6
##   AirQualityStationEoI~1 AirQualityStationName AveragingTime DatetimeBegin
##   <chr>                  <chr>                 <chr>         <dttm>
## 1 IT0477A                MILANO - V.LE MARCHE  day           2022-01-01 01:00:00
## 2 IT0477A                MILANO - V.LE MARCHE  day           2022-01-02 01:00:00
## 3 IT0477A                MILANO - V.LE MARCHE  day           2022-01-03 01:00:00
## 4 IT0477A                MILANO - V.LE MARCHE  day           2022-01-04 01:00:00
## 5 IT0477A                MILANO - V.LE MARCHE  day           2022-01-05 01:00:00
## 6 IT0477A                MILANO - V.LE MARCHE  day           2022-01-06 01:00:00
## # i abbreviated name: 1: AirQualityStationEoICode
## # i 2 more variables: DatetimeEnd <dttm>, PM10 <dbl>
```

Retrieve NO_2 data for a specific macroregion (Eurostat classification NUTS-1) given its name `LATN_NAME`

```
# Identify the names of the areas from which to download the data
zones <- c("Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest","Vlaams Gewest","West-Nederlan

# Download the corresponding data
data <- EEAaq_get_data(
  zone_name = zones,                # LAU zone code
  NUTS_level = "NUTS1",             # NUTS level
  pollutants = c("NO2", "PM10"),    # Pollutant
  from = "2023-01-01",              # Start date
  to = "2023-12-31",                # End date
  verbose = FALSE                   # Print detailed progress
)
```

```
unique(data$AirQualityStationEoICode)
```

```
##  [1] "BELAL01" "BELAT83" "BELHB23" "BETB001" "BETB004" "BETB006" "BETB008"
##  [8] "BETB011" "BETBUL1" "BETCHA1" "BETE013" "BETE714" "BETE716" "BETM802"
## [15] "BETMEU1" "BETN043" "BETR001" "BETR002" "BETR012" "BETR701" "BETR702"
## [22] "BETR721" "BETR740" "BETR801" "BETR802" "BETR803" "BETR804" "BETR805"
## [29] "BETR806" "BETR817" "BETR818" "BETR822" "BETR831" "BETR842" "BETR891"
## [36] "BETR897" "BETREG1" "BETVBX1" "BETVBX2" "BETVBX3" "NL00136" "NL00138"
## [43] "NL00236" "NL00237" "NL00240" "NL00241" "NL00247" "NL00546" "NL00551"
## [50] "NL00553" "NL00556" "NL00570" "NL00572" "NL00573" "NL00701" "NL00704"
```

Note 1: If the query's zone_name parameter corresponds to a valid *CITY_NAME* (i.e., not NULL in the dataset), the function will return the corresponding data. If no valid CITY_NAME is associated with the zone_name, the function attempts to retrieve all available data for the entire country and subsequently filter for the specified zone_name.

Note 2: For very small towns or certain countries such as Turkey or Albania, data may not currently be available in the dataset.This limitation reflects the data unavailability at the EEA Air Quality Viewer.

Note 3: If the parameters used in the query include *polygon* or *quadrant*, the function outputs an EEAaq_df_sfc object. Otherwise, it returns an EEAaq_df object, which is a tibble dataframe.

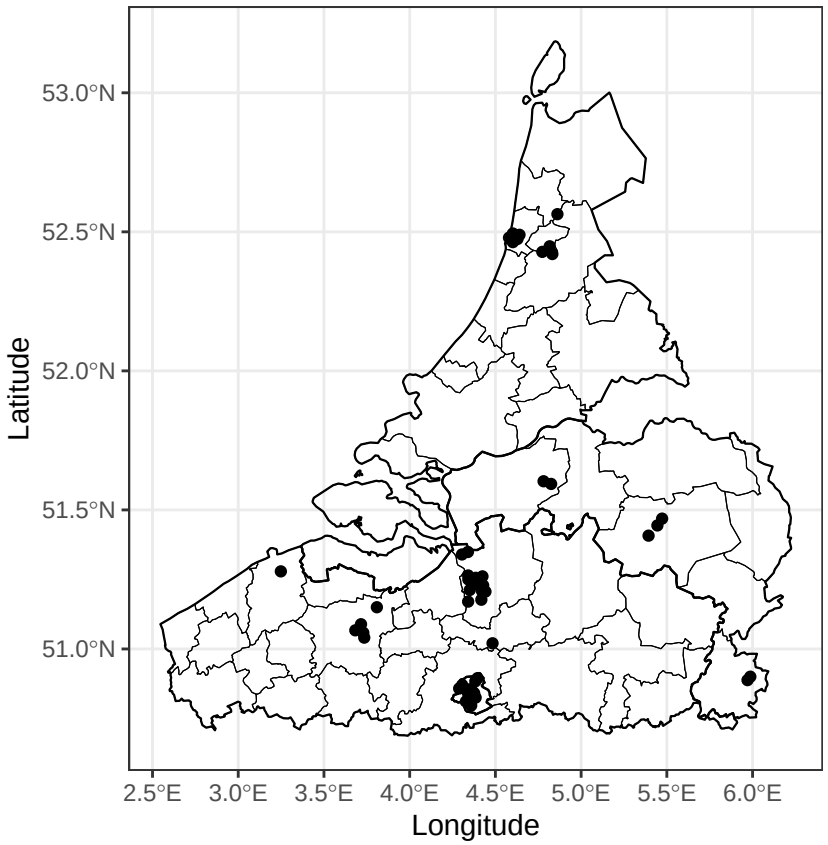
EEAaq map stations

`EEAaq_map_stations` generates a static or dynamic map of user-defined monitoring stations. The function accepts as input either an object of the `EEAaq_df` class (default output of the `EEAaq_get_data` function), or all other parameters specifying the area and the pollutants.

Map the stations using as `EEAaq_df` object, the dataset concerning NO_2 and PM_{10} in Belgium and The Netherlands

```
EEAaq_map_stations(
  data = data,
  bounds_level = "NUTS3",
  color = FALSE,
  dynamic = FALSE
)
```

```
## Simple feature collection with 4 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 2.546088 ymin: 50.688 xmax: 6.225231 ymax: 53.18511
## Geodetic CRS:  WGS 84
##   NUTS_ID LEVL_CODE CNTR_CODE
## 1     BE1         1        BE
## 2     BE2         1        BE
## 3     NL3         1        NL
## 4     NL4         1        NL
##                                                  NAME_LATN
## 1 Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest
## 2                                          Vlaams Gewest
## 3                                         West-Nederland
## 4                                         Zuid-Nederland
##                                                  NUTS_NAME MOUNT_TYPE
## 1 Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest         NA
## 2                                          Vlaams Gewest         NA
## 3                                         West-Nederland         NA
## 4                                         Zuid-Nederland         NA
##   URBN_TYPE COAST_TYPE                       geometry
## 1        NA         NA MULTIPOLYGON (((4.415738 50...
## 2        NA         NA MULTIPOLYGON (((5.776583 50...
## 3        NA         NA MULTIPOLYGON (((5.171192 52...
## 4        NA         NA MULTIPOLYGON (((5.518671 51...
## points Country ISO AirQualityStationEoICode AirQualityStationNatCode AirQualityStationName Altitude
```



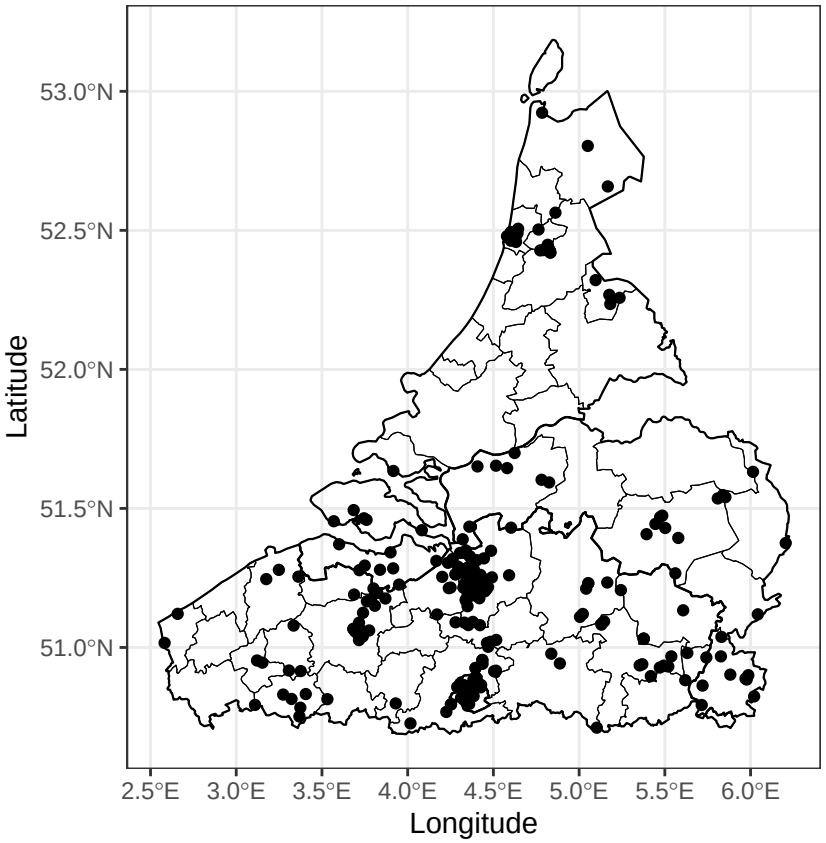
Note: Using the parameter `bounds_level = "NUTS3"`, the map is generated with internal boundaries corresponding to the NUTS-3 level. The same output could be obtained specifying explicitly the zone information.

Map all the stations monitoring NO_2 and PM_{10} in Belgium and The Netherlands

```
EEAaq_map_stations(
  zone_name = zones,
  NUTS_level = "NUTS1",
  pollutant = c("NO2", "PM10"),
  bounds_level = "NUTS3",
  color = FALSE,
  dynamic = FALSE
)
```

```
## Simple feature collection with 4 features and 8 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 2.546088 ymin: 50.688 xmax: 6.225231 ymax: 53.18511
## Geodetic CRS:  WGS 84
##   NUTS_ID LEVL_CODE CNTR_CODE
## 1     BE1         1        BE
## 2     BE2         1        BE
## 3     NL3         1        NL
## 4     NL4         1        NL
##                                                    NAME_LATN
## 1 Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest
```

```
## 2                                                Vlaams Gewest
## 3                                               West-Nederland
## 4                                               Zuid-Nederland
##                                                     NUTS_NAME MOUNT_TYPE
## 1 Région de Bruxelles-Capitale/Brussels Hoofdstedelijk Gewest         NA
## 2                                                Vlaams Gewest         NA
## 3                                               West-Nederland         NA
## 4                                               Zuid-Nederland         NA
##   URBN_TYPE COAST_TYPE                          geometry
## 1        NA         NA MULTIPOLYGON (((4.415738 50...
## 2        NA         NA MULTIPOLYGON (((5.776583 50...
## 3        NA         NA MULTIPOLYGON (((5.171192 52...
## 4        NA         NA MULTIPOLYGON (((5.518671 51...
## points Country ISO AirQualityStationEoICode AirQualityStationNatCode AirQualityStationName Altitude
```



EEAaq summary

This function aims to describe the dataset that has been previously imported, both at a global level, which means considering the complete set of time stamps and monitoring stations in the dataset, and at the station-specific level, where summary statistics and information are grouped by monitoring station.
In addition to basic exploratory descriptive statistics (e.g., average pollutant concentration, variability, measures of skewness and kurtosis), the function provides information about the gap length and the correlation between pollutants if at least two pollutants are considered simultaneously.
The `EEAaq_summary` function receives as input an `EEAaq_df` object, i.e. the output of the EEAaq get data function.

Compute the descriptive statistics

```
summ <- EEAaq_summary(data = data)
```

```
## The dataset contains:
##  ** 477510 total observations
##  ** 56 stations
##  ** 8736 time stamps: from 2023-01-01 01:00:00 to 2023-12-31
```

Print screen the global statsitics

```
summ$Summary
```

```
## # A tibble: 2 x 8
##   Pollutant NA_count NA_perc negative_count   min  mean   max    sd
##   <chr>        <int>   <dbl>          <int> <dbl> <dbl> <dbl> <dbl>
## 1 PM10        162930    34.1           3465     0  18.0 1565.  12.5
## 2 NO2          50492    10.6             24     0  20.1  299   14.3
```

Print screen the station-specific statsitics

```
summ$Summary_byStat$Mean_byStat
```

```
## # A tibble: 56 x 4
##    AirQualityStationEoICode AirQualityStationName             PM10   NO2
##    <chr>                    <chr>                            <dbl> <dbl>
##  1 BELAL01                  40AL01 - ANTWERPEN                18.0  20.1
##  2 BELAT83                  40AT83 - BERENDRECHT              18.0  20.1
##  3 BELHB23                  40HB23 - HOBOKEN                  18.0  20.1
##  4 BETB001                  41B001 - BRUSSEL (Kunst-Wet)      18.0  20.1
##  5 BETB004                  41B004 - STE.CATHERI              18.0  20.1
##  6 BETB006                  41B006 - PARL.EUROPE              18.0  20.1
##  7 BETB008                  41B008 - Brussel (Beliardstraat)  18.0  20.1
##  8 BETB011                  41B011 - BERCHEM S.A              18.0  20.1
##  9 BETBUL1                  41BUL1 - BRUXELLES                18.0  20.1
## 10 BETCHA1                  41CHA1 - GANSHOREN                18.0  20.1
## # i 46 more rows
```

Print screen the linear correlation matrix

```
summ$Corr_Matrix
```

```
## # A tibble: 56 x 4
##    AirQualityStationEoICode AirQualityStationName            PM10_NO2 NO2_PM10
##    <chr>                    <chr>                               <dbl>    <dbl>
##  1 BELAL01                  40AL01 - ANTWERPEN                  0.431    0.431
##  2 BELAT83                  40AT83 - BERENDRECHT                0.152    0.152
##  3 BELHB23                  40HB23 - HOBOKEN                   NA       NA
##  4 BETB001                  41B001 - BRUSSEL (Kunst-Wet)       NA       NA
##  5 BETB004                  41B004 - STE.CATHERI               NA       NA
##  6 BETB006                  41B006 - PARL.EUROPE               NA       NA
##  7 BETB008                  41B008 - Brussel (Beliardstraat)   NA       NA
##  8 BETB011                  41B011 - BERCHEM S.A                0.495    0.495
##  9 BETBUL1                  41BUL1 - BRUXELLES                 NA       NA
```

```
## 10 BETCHA1                  41CHA1 - GANSHOREN                NA       NA
## # i 46 more rows
```

EEAaq time aggregate

Recall that most pollutants are monitored by EEA on a hourly or daily basis, posing challenges for interpretation and representation. The `EEAaq_time_aggregate` function simplifies this by aggregating data into annual, monthly, weekly, daily, or hourly intervals, generating summary statistics for each station in an `EEAaq_taggr_df` object.

Get the station-specific monthly minimum, maximum, average and median concentrations of NO_2 and PM_{10} in Belgium and The Netherlands

```
t_aggr <- EEAaq_time_aggregate(
  data = data,
  frequency = "monthly",
  aggr_fun = c("min", "max", "mean", "median" )
)
```

Print screen of the aggregated (monthly) data

```
t_aggr$TimeAggr
```

```
## # A tibble: 668 x 11
##    AirQualityStationEoICode AirQualityStationName Date       PM10_min PM10_max
##    <chr>                    <chr>                 <date>        <dbl>    <dbl>
##  1 BELAL01                  40AL01 - ANTWERPEN    2023-01-01      3.9     77.4
##  2 BELAL01                  40AL01 - ANTWERPEN    2023-02-01      5.4     76.4
##  3 BELAL01                  40AL01 - ANTWERPEN    2023-03-01      4.4     87.4
##  4 BELAL01                  40AL01 - ANTWERPEN    2023-04-01      3.9     82.9
##  5 BELAL01                  40AL01 - ANTWERPEN    2023-05-01      8.4    165.
##  6 BELAL01                  40AL01 - ANTWERPEN    2023-06-01      6.9     65.9
##  7 BELAL01                  40AL01 - ANTWERPEN    2023-07-01      4.9     49.9
##  8 BELAL01                  40AL01 - ANTWERPEN    2023-08-01      5.4     54.4
##  9 BELAL01                  40AL01 - ANTWERPEN    2023-09-01      5.4    115.
## 10 BELAL01                  40AL01 - ANTWERPEN    2023-10-01      4.9     51.4
## # i 658 more rows
## # i 6 more variables: PM10_mean <dbl>, PM10_median <dbl>, NO2_min <dbl>,
## #   NO2_max <dbl>, NO2_mean <dbl>, NO2_median <dbl>
```

Print screen of the PM_{10} aggregated data only

```
t_aggr$TimeAggr_byPollutant$PM10
```

```
## # A tibble: 668 x 7
##    AirQualityStationEoICode AirQualityStationName Date         min   max  mean
##    <chr>                    <chr>                 <date>     <dbl> <dbl> <dbl>
##  1 BELAL01                  40AL01 - ANTWERPEN    2023-01-01   3.9  77.4  19.1
##  2 BELAL01                  40AL01 - ANTWERPEN    2023-02-01   5.4  76.4  27.3
##  3 BELAL01                  40AL01 - ANTWERPEN    2023-03-01   4.4  87.4  16.1
##  4 BELAL01                  40AL01 - ANTWERPEN    2023-04-01   3.9  82.9  20.1
##  5 BELAL01                  40AL01 - ANTWERPEN    2023-05-01   8.4 165.   24.5
##  6 BELAL01                  40AL01 - ANTWERPEN    2023-06-01   6.9  65.9  24.9
```

```
##  7 BELAL01                  40AL01 - ANTWERPEN    2023-07-01   4.9  49.9  14.8
##  8 BELAL01                  40AL01 - ANTWERPEN    2023-08-01   5.4  54.4  15.3
##  9 BELAL01                  40AL01 - ANTWERPEN    2023-09-01   5.4 115.   20.6
## 10 BELAL01                  40AL01 - ANTWERPEN    2023-10-01   4.9  51.4  17.3
## # i 658 more rows
## # i 1 more variable: median <dbl>
```

EEAaq_idw_map

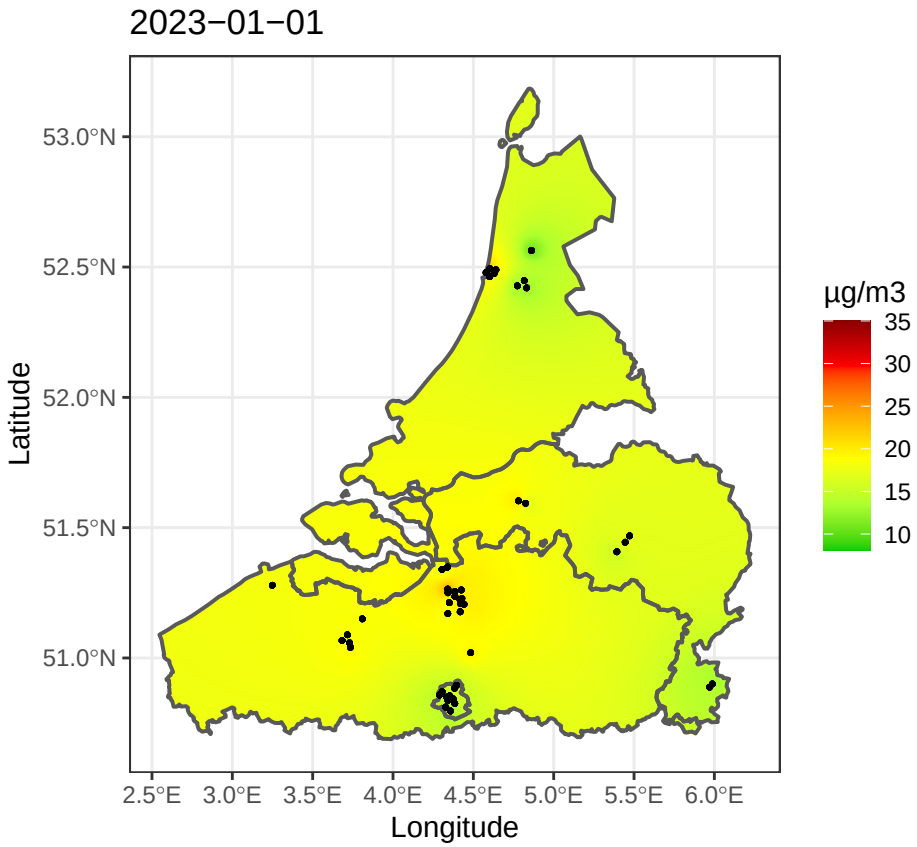
To enable quick and intuitive visual analysis, the `EEAaq_idw_map` function provides spatial interpolation maps using the Inverse Distance Weighting (IDW) method (Shepard, 1968). This technique estimates the value of a variable at unknown locations by calculating a weighted average of known values, with weights inversely proportional to the distance from known points. Closer points contribute more heavily to the estimate, making it a practical approach for interpolating geolocated air quality data.

Generate IDW interpolated maps of monthly average concentrations of NO_2 in the Netherlands and Belgium

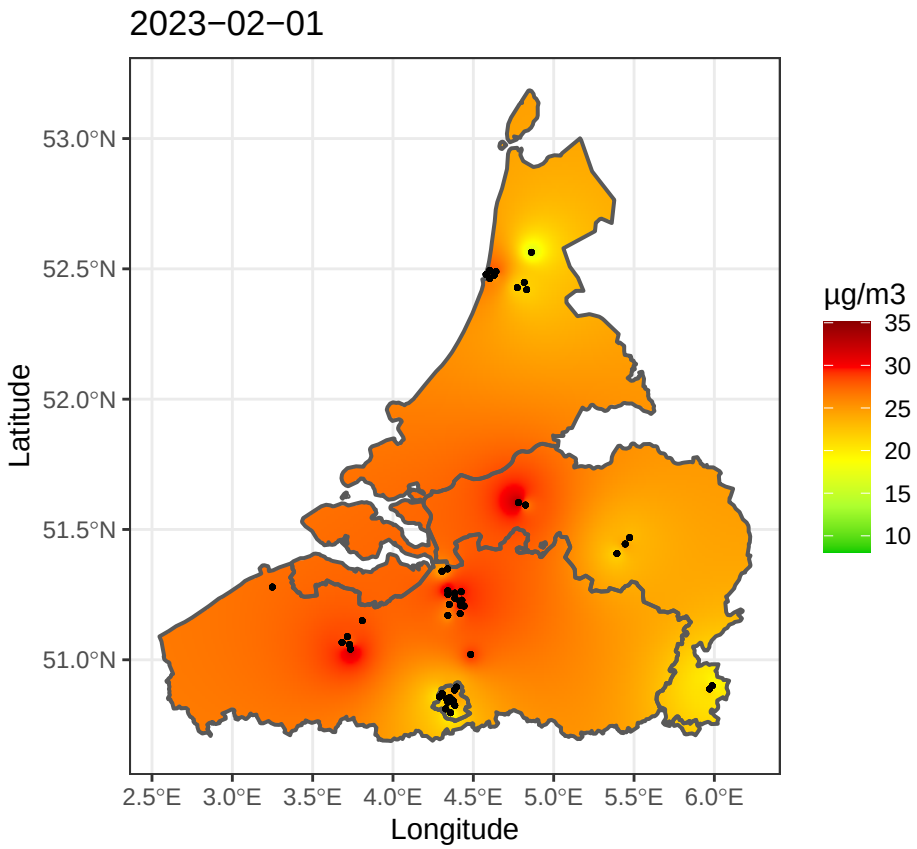
```
EEAaq::EEAaq_idw_map(
  data = t_aggr,
  pollutant = "PM10",
  aggr_fun = "mean",
  distinct = TRUE,
  gradient = TRUE,
  idp = 2
)
```

```
## Map initialization started at 2025-02-05 13:42:48.973053
## Map initialization ended at 2025-02-05 13:43:00.499283
## Computing IDW interpolation started at 2025-02-05 13:43:00.499448
## Computing IDW interpolation for: 2023-01-01, 1 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-02-01, 2 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-03-01, 3 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-04-01, 4 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-05-01, 5 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-06-01, 6 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-07-01, 7 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-08-01, 8 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-09-01, 9 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-10-01, 10 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-11-01, 11 of 12
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-12-01, 12 of 12
## [inverse distance weighted interpolation]
```

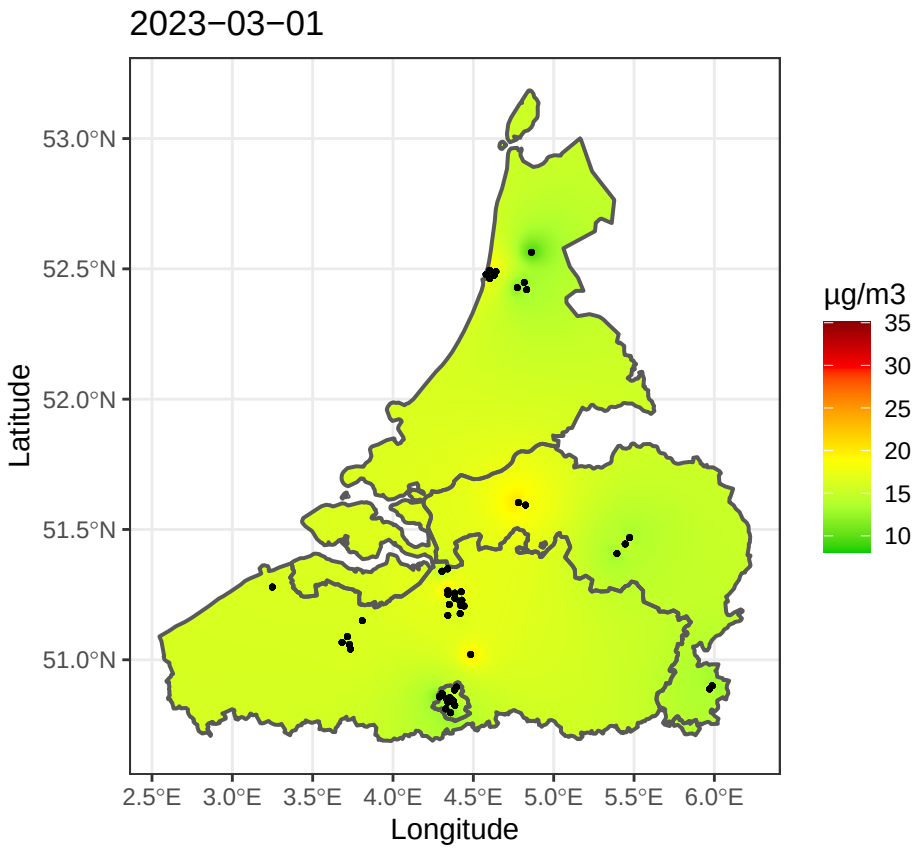
```
## Computing IDW interpolation ended at 2025-02-05 13:44:33.663341
## [[1]]
```



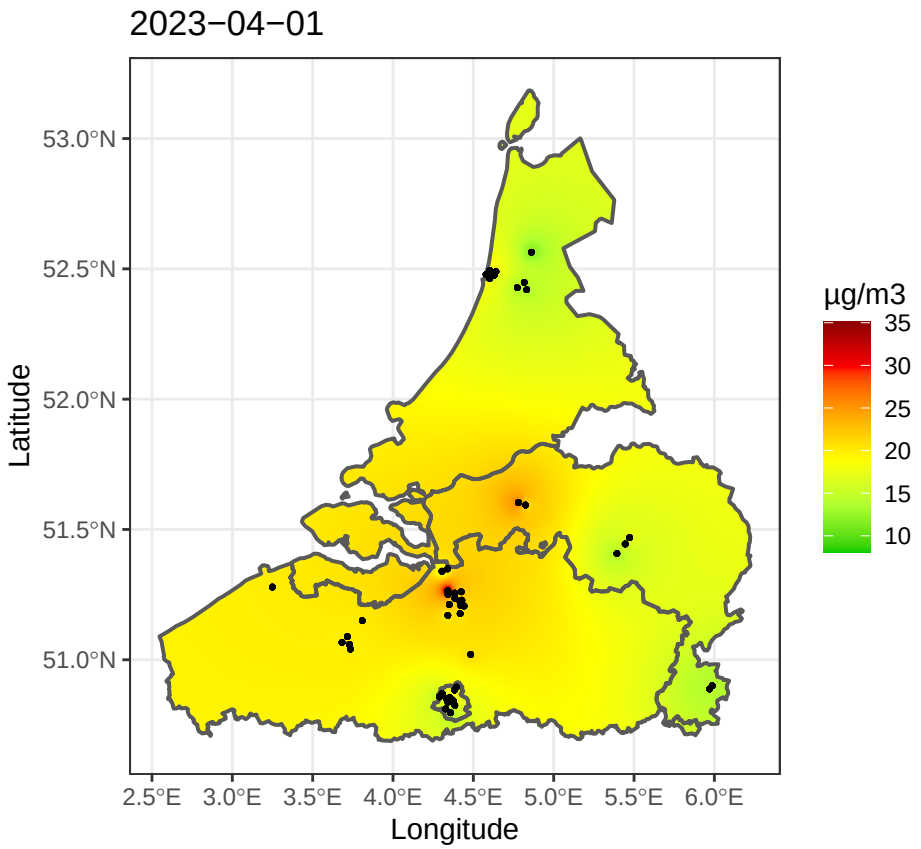

```
##
## [[2]]
```


2023−02−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

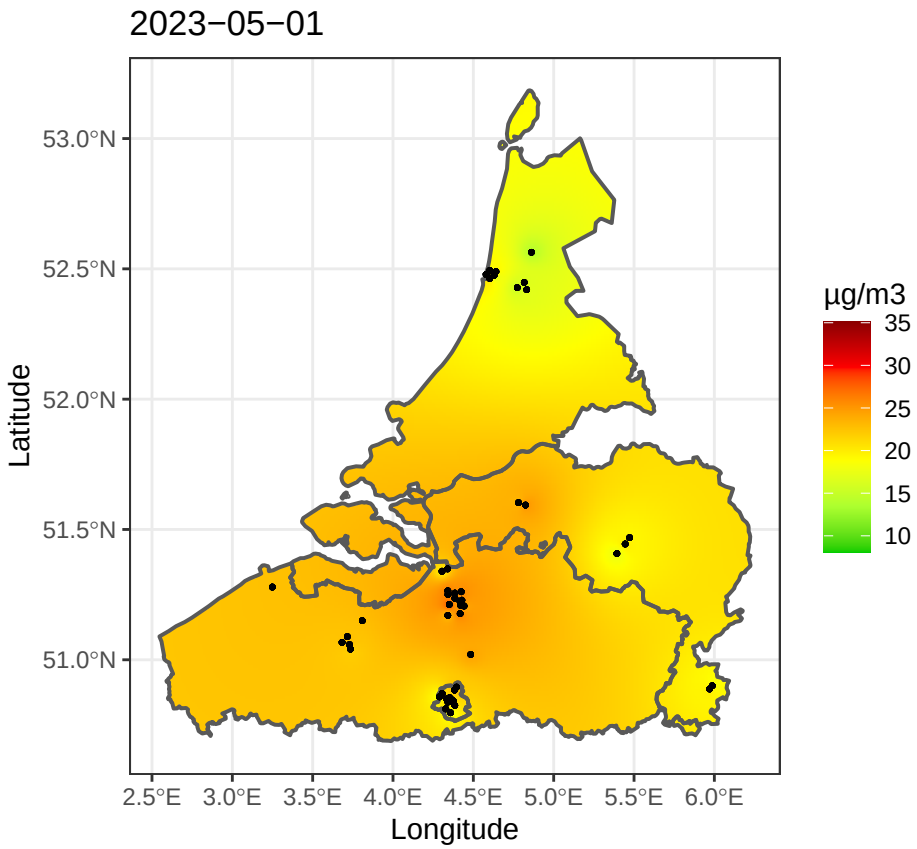
```
##
## [[3]]
```


2023−03−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

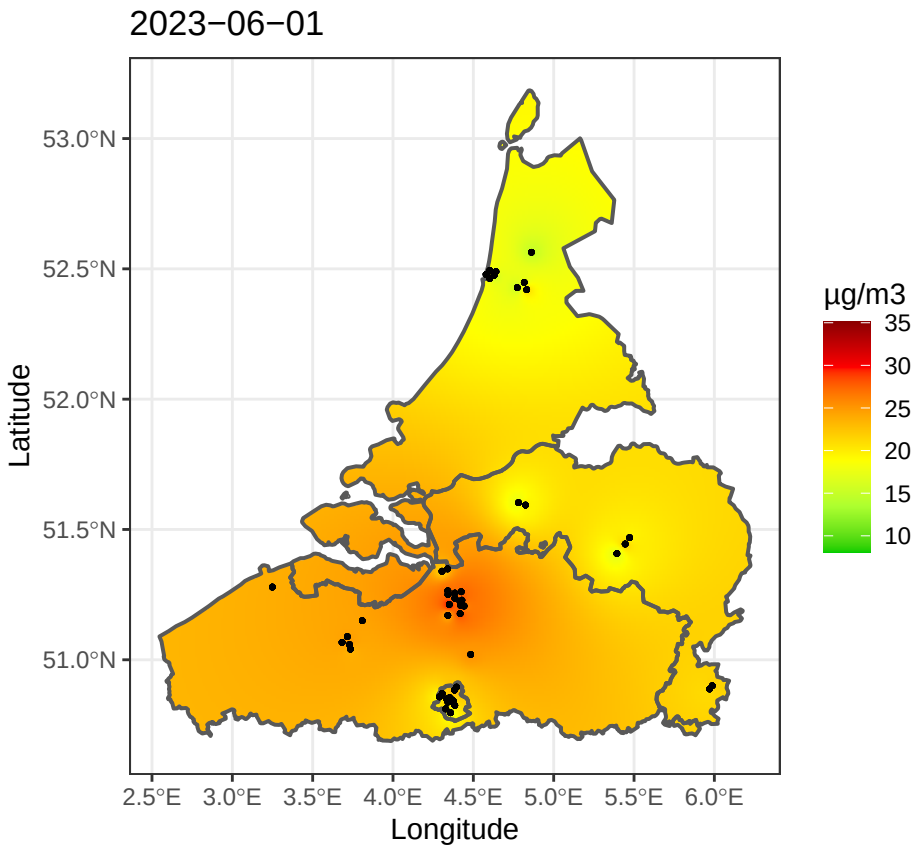
```
##
## [[4]]
```


2023−04−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

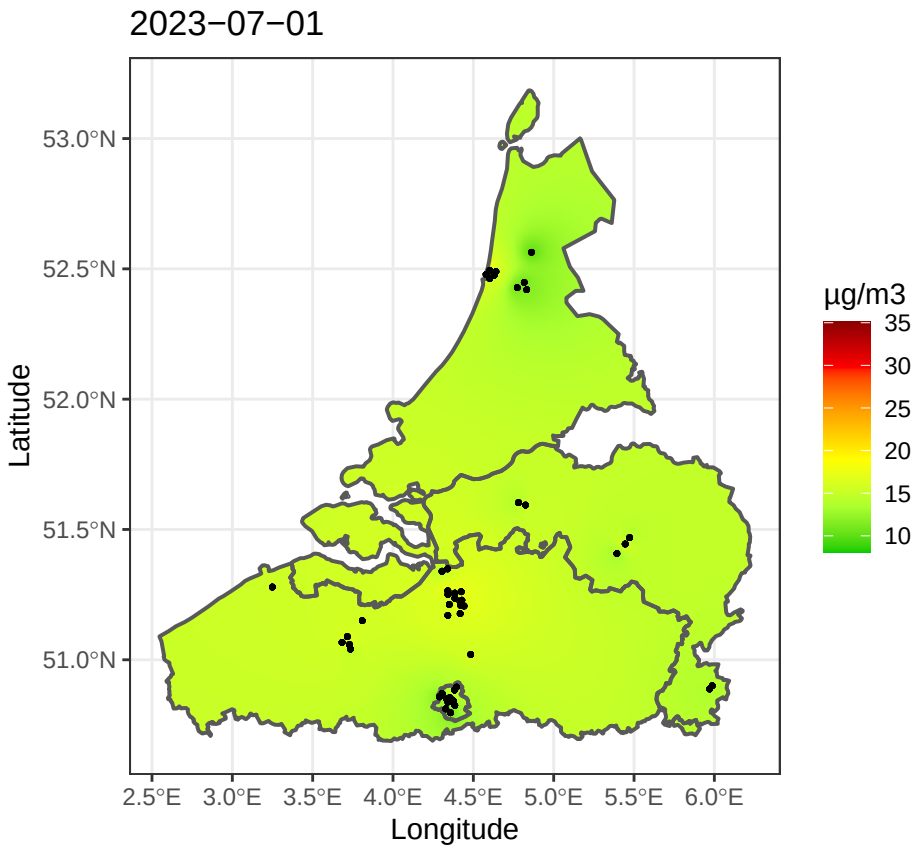
```
##
## [[5]]
```


2023−05−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

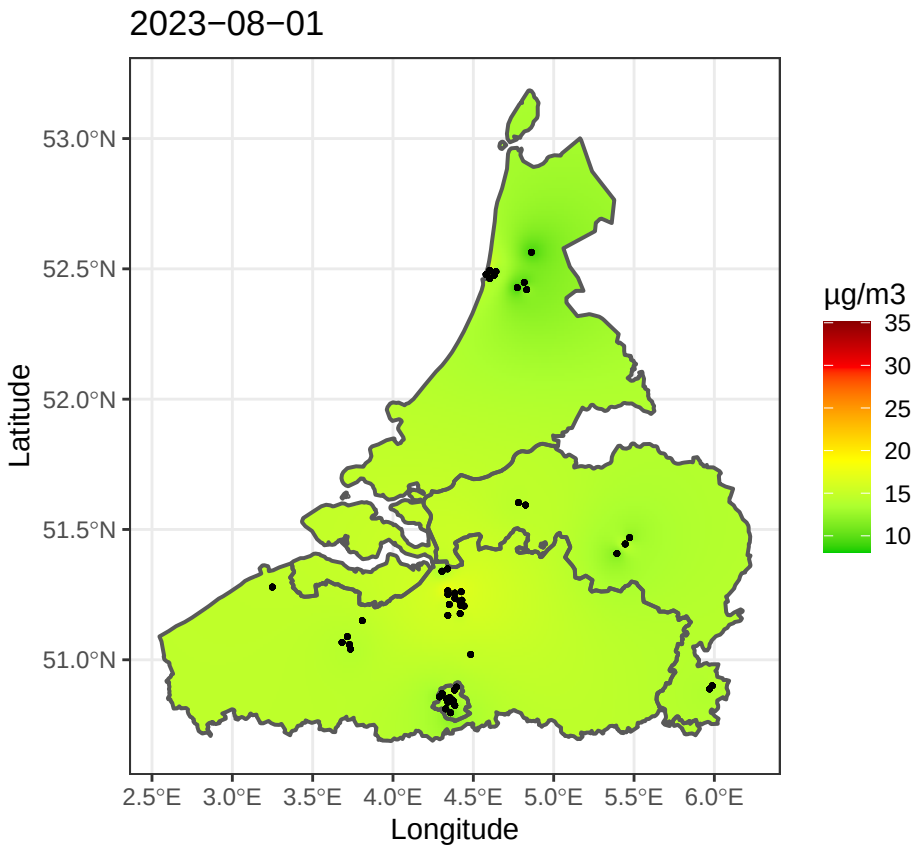
```
##
## [[6]]
```


2023−06−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

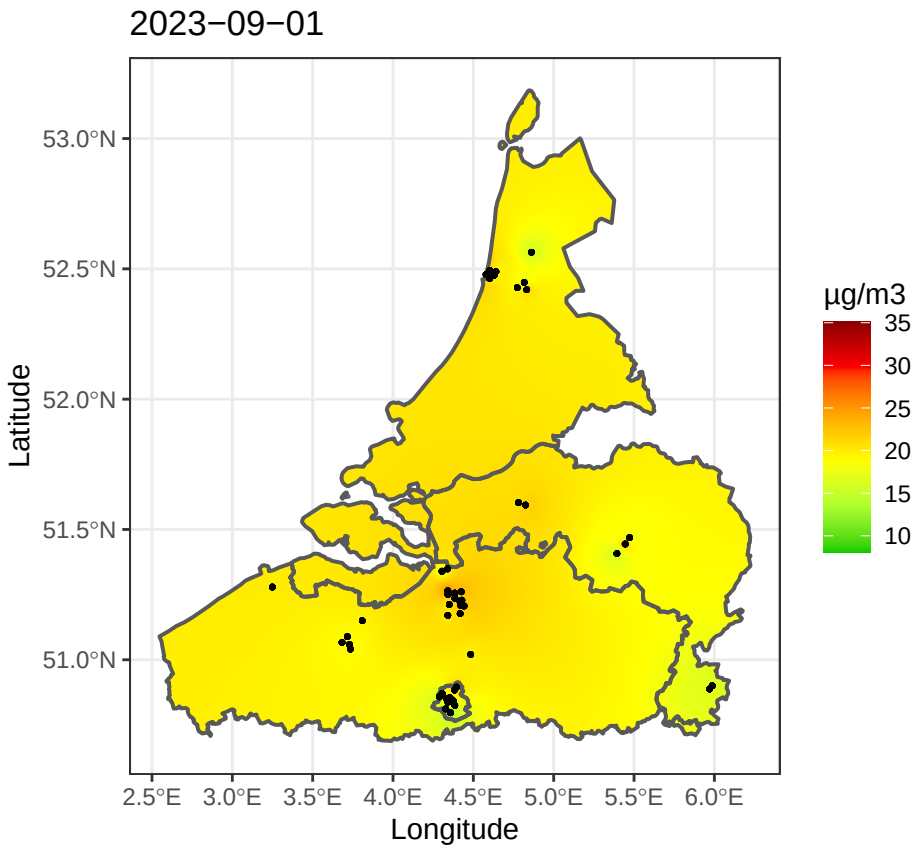
```
##
## [[7]]
```


2023−07−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

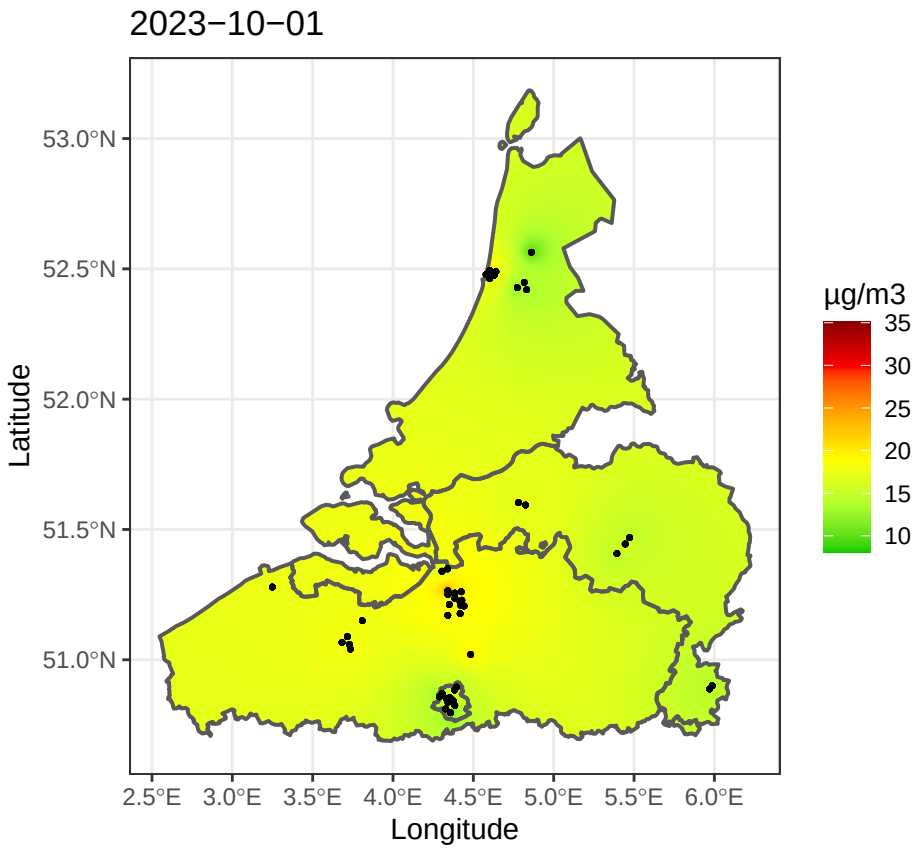
```
##
## [[8]]
```


2023−08−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

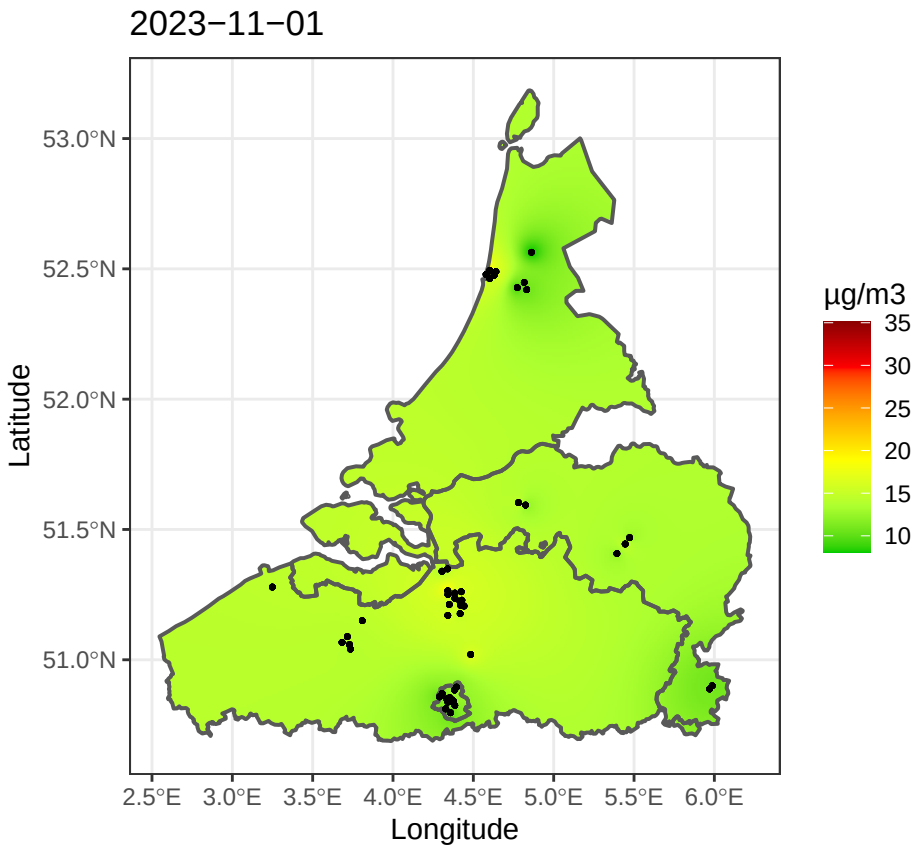
```
##
## [[9]]
```


2023−09−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

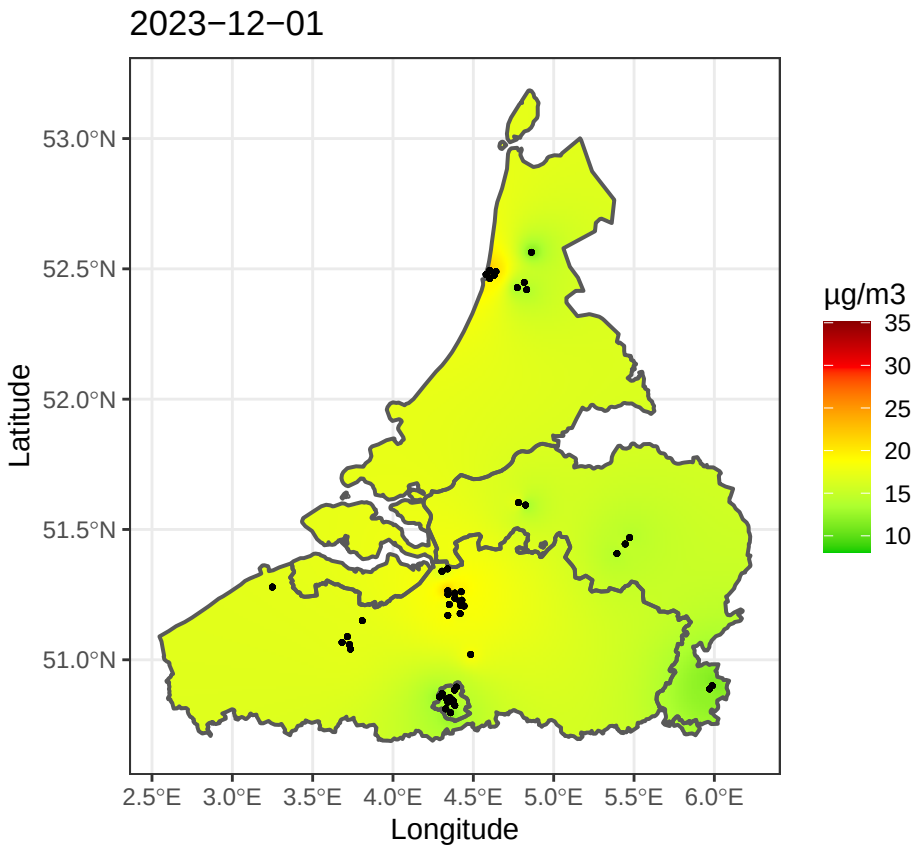
```
##
## [[10]]
```


2023−10−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

```
##
## [[11]]
```


2023−11−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
35
30
25
20
15
10

```
##
## [[12]]
```

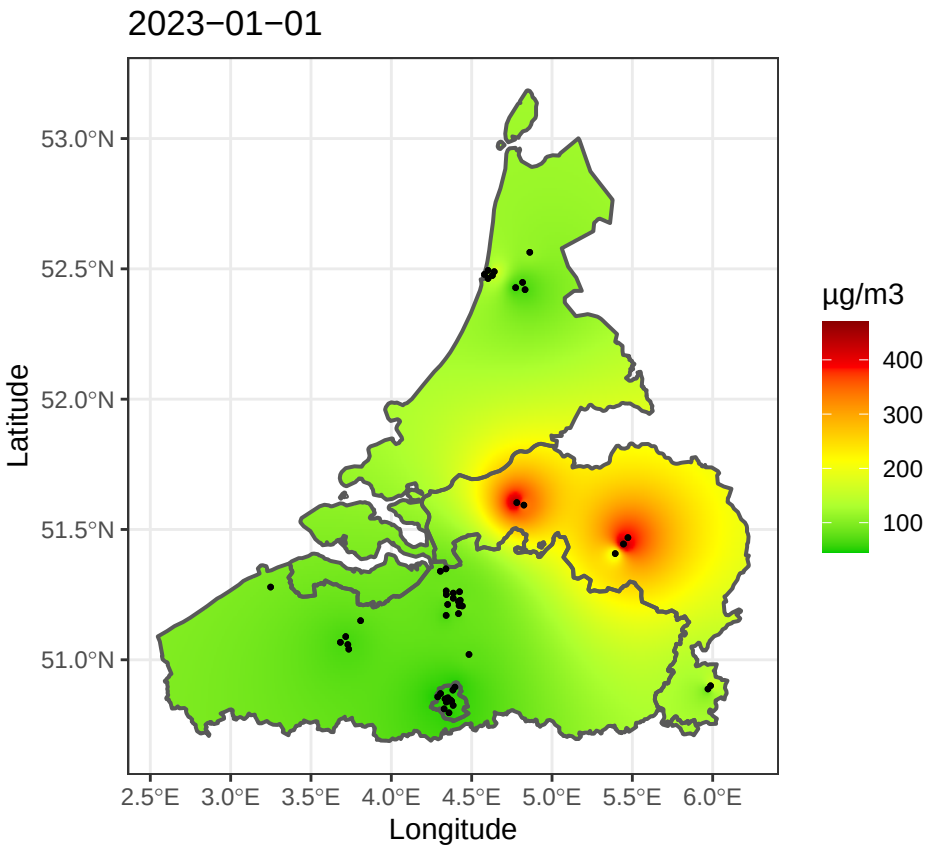


Generate IDW interpolated maps of the maximum monthly concentrations of NO_2 in january and february 2023 in the Netherlands and Belgium

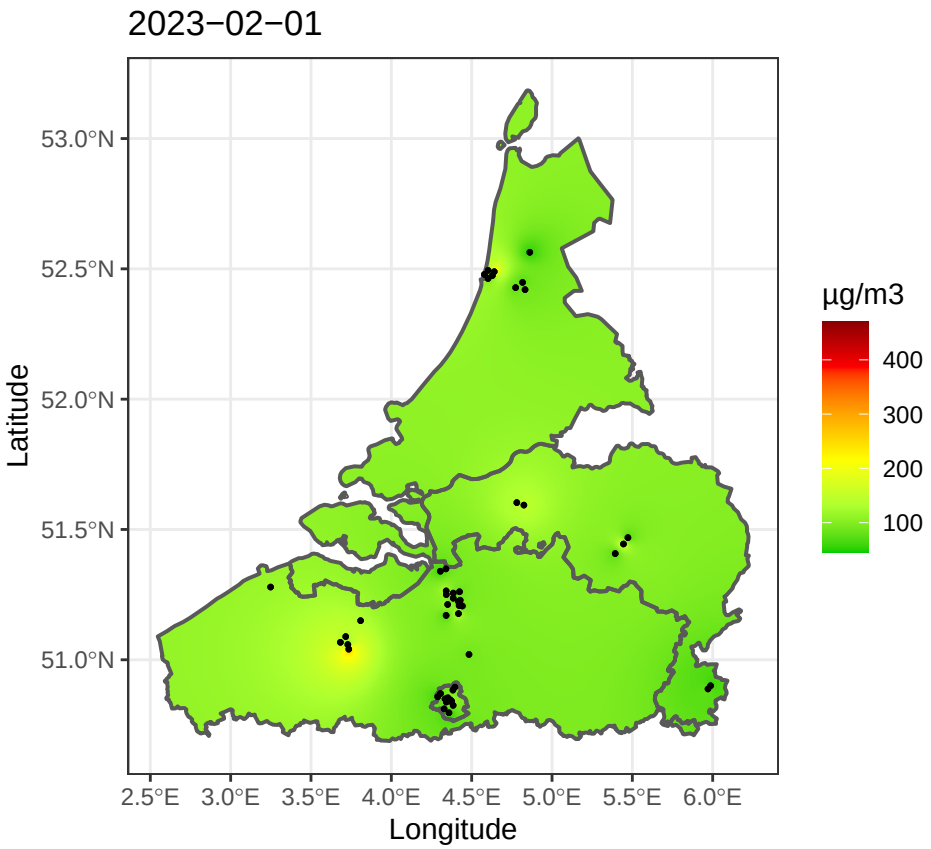
```
EEAaq::EEAaq_idw_map(
  data = t_aggr$TimeAggr_byPollutant$PM10 %>% dplyr::filter(Date %in% c("2023-01-01","2023-02-01")),
  pollutant = "PM10",
  aggr_fun = "max",
  distinct = TRUE,
  gradient = TRUE,
  idp = 2
)
```

```
## Map initialization started at 2025-02-05 13:44:37.125827
## Map initialization ended at 2025-02-05 13:44:49.815679
## Computing IDW interpolation started at 2025-02-05 13:44:49.815864
## Computing IDW interpolation for: 2023-01-01, 1 of 2
## [inverse distance weighted interpolation]
## Computing IDW interpolation for: 2023-02-01, 2 of 2
## [inverse distance weighted interpolation]
## Computing IDW interpolation ended at 2025-02-05 13:45:04.718012
```

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


2023−01−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
400
300
200
100

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


2023−02−01
53.0°N
52.5°N
52.0°N
51.5°N
51.0°N
Latitude
2.5°E
3.0°E
3.5°E
4.0°E
4.5°E
5.0°E
5.5°E
6.0°E
Longitude
µg/m3
400
300
200
100