More information

18/05/2023

Contents

I've tried to add more detail because whilst the problems all have some similarities (they're all simulations using a computer model), there are different things you can do with it. Some are simulating events that have already happened and we're trying to learn something about this; some are simulating hypothetical events that haven't happened, but that are important to understand.

That said, you could use any of these datasets in other ways. E.g. for the eruptions/chemical releases, use an out-of-sample simulation as the 'truth'. Or use the Covid model to assess things like 'what would have happened at the start of the Covid pandemic in the UK if the R number was 0.5 higher?' - so now studying some hypothetical pandemic.

Dispersion simulations (volcanic ash, radiological releases)

These are SIMULATIONS - not real eruptions/releases, so there's no real 'observations' to compare to - but at some point events like this will happen, and want to be able to predict what will happen. Or, assess effects of events like these in advance. Useful for risk/insurance industries, also for aviation.

In the data here https://doi.org/10.5281/zenodo.4770066 there's

a) simulated eruptions of Hekla (an Icelandic volcano)
b) simulated eruptions of Öræfajökull (a different Icelandic volcano)
c) simulated radiological release from 12 locations in Europe.

It is enough to only consider ONE of these - each has enough simulations (around 240) and therefore data.

These are all simulations - not actually happened - being performed with different weather patterns.

Chemical releases - again, simulations, but from 12 different locations. Again the difference is in the weather pattern.

There are some similarities across all of these:

- the only input that is changing is the weather - each time the eruption/chemical release is the same, but where the ash/chemicals end up is different
- the different simulations are labelled by time (relating to the weather being used to simulate the spread)

Some differences:

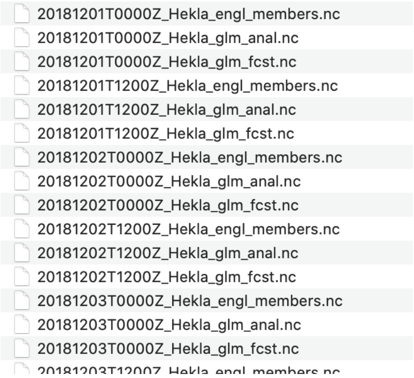
- the volcano data gives ash every 3 hours, for 24 hours
- the radiological data just gives caesium deposition after 48 hours
- the volcano data also gives ash at different levels in the atmosphere ('flight levels')

So there's different characteristics and different things that can be explored - we can't look at how the chemical cloud evolves over time, only its extent after 48 hours.

A subtlety:

- within each individual simulation, there are 3 different things:

1) an ‘analysis’ scenario (tag `glm_anal`)
2) a ‘deterministic forecast’ (tag `glm_fcst`)
3) 18 ‘probabilistic forecasts’ (tag `engl_members`)



```
20181201T0000Z_Hekla_engl_members.nc
20181201T0000Z_Hekla_glm_anal.nc
20181201T0000Z_Hekla_glm_fcst.nc
20181201T1200Z_Hekla_engl_members.nc
20181201T1200Z_Hekla_glm_anal.nc
20181201T1200Z_Hekla_glm_fcst.nc
20181202T0000Z_Hekla_engl_members.nc
20181202T0000Z_Hekla_glm_anal.nc
20181202T0000Z_Hekla_glm_fcst.nc
20181202T1200Z_Hekla_engl_members.nc
20181202T1200Z_Hekla_glm_anal.nc
20181202T1200Z_Hekla_glm_fcst.nc
20181203T0000Z_Hekla_engl_members.nc
20181203T0000Z_Hekla_glm_anal.nc
20181203T0000Z_Hekla_glm_fcst.nc
20181203T1200Z_Hekla_engl_members.nc
```

The reason for this is that the original paper was comparing the deterministic and probabilistic forecasts with each other, and used the ‘analysis’ as the ’truth’ they were comparing to.

It’s possible to use all 3 of these sources of data at each time point, depending on what you’re trying to do.

However, if you want to link the output to the input weather in some way, you can only use the ‘analysis’ versions (this is because the weather files themselves are huge, and weren’t stored for every forecast, whereas the analysis version is stored anyway). i.e. we don’t know the input weather for the deterministic and probabilistic forecasts. If doing something else, then can use whichever.

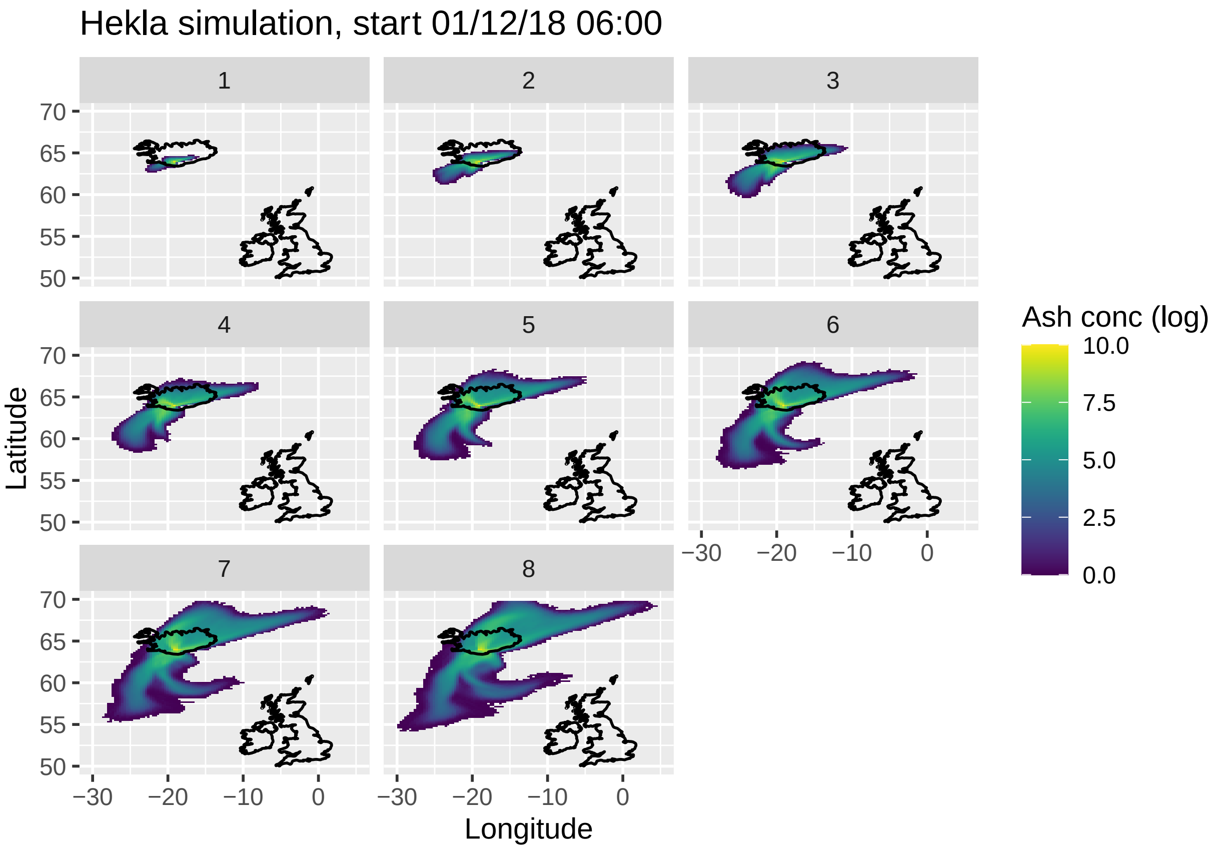
(An interesting use of these deterministic/probabilistic forecasts could be to treat them as ‘observations’, and try to learn what the input weather was that generated them.)

The weather data

Is stored on JASMIN, I’m currently getting permission to access it, then I’ll process it a bit (it’s probably very large) and share it.

Hekla example

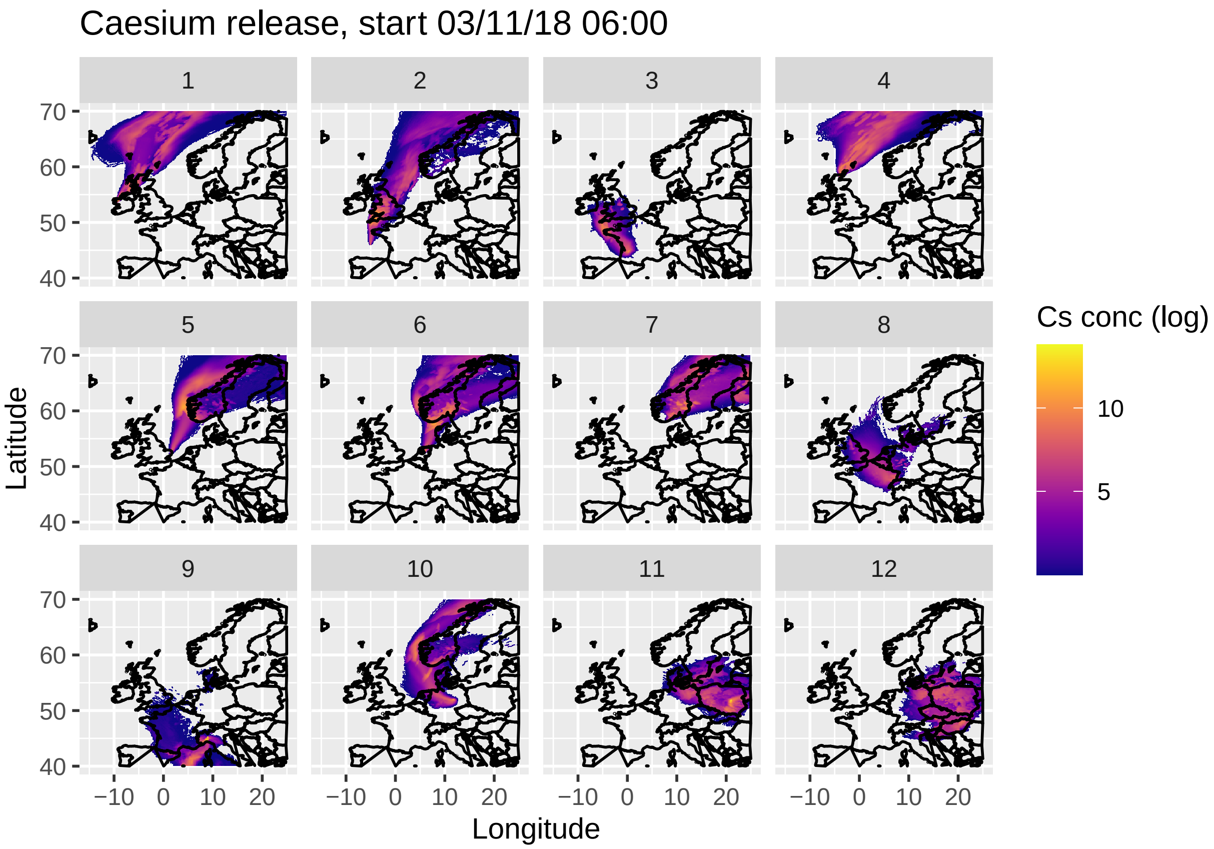
Plot of analysis simulation initialised at 01/12/18 06:00: each panel shows a 3 hour average, for the 1st 24 hours of the eruption, of ash concentration (log) integrated vertically (i.e. not for a particular flight level, but a total):



Instead, for this particularly simulation you could plot the surface level ash (linking to human imapcts?) or for different flight levels (linking to aviation impacts?). And you could do this for all 240 different simulations!

Caesium releases

In each file here, there are releases of Caesium-137 from 12 different locations. Each simulation of the dispersion model is done independently for each location, but then the output from the 12 different simulations, for the chosen date/weather, is stacked together in a single file, and I've plotted a single file below, showing Caesium deposition after 48 hours:



Some of the release locations are in the UK, some elsewhere in Europe.

As with the eruptions, there are around 240 simulations **from each location**, and you could just select a single location to study.

There's probably a much clearer link in to health risk/impact here, as most of these releases are happening nearer to large populations, as opposed to the Icelandic volcanoes where a lot of the time the ash plumes go out over the ocean.

Covid

This model is definitely not perfect, however it should be able to match the observations in some way, and understanding where it fails, and how its estimates differ from other models, is useful. There are lots of different Covid models out there, and from these you will usually get different estimated 'best' parameters, because of the assumptions that have been made whilst building the Covid model (it's very hard to simulate individuals moving around and all their contacts, so simplifications have to be made).

This is different from the dispersion simulations as this is attempting to replicate what actually happened in the Covid-19 pandemic in the UK (with the idea that in a future pandemic, we will therefore understand a lot more how to model this, how to calibrate unknown parameters, how to better understanding uncertainty in modelling and decision making). Hence, there are observations that we can compare the different simulations to.

Goal: find settings of the input parameters (R0, probability of hospitalisation, etc.) that lead to simulation output consistent (in some sense) with what actually happened.

These simulations are up to the start of lockdown 1 (at this point, movements/contacts etc. change, and this fundamentally changes the model. Want to understand the parameters governing the dynamics of Covid **before** these changes take effect and make it harder to estimate these).

The data is available on dropbox:

- `inputs.csv` found here: https://www.dropbox.com/s/s0holoyxt7ldecp/inputs.csv?dl=0
- `outputs.rds` found here: https://www.dropbox.com/s/uvjqppch0l596wf/outputs.rds?dl=0

Each row of `inputs` is a different set of input parameters for the COVID simulator, e.g. row 1 is input vector 1:

```
inputs[1,]
```

```
##         R0        TE       TP      TI1        TI2       nuA       ns
## 1 3.619941 0.8965843 2.012114 4.256211 0.08660611 0.9108957 43.22405
##   p_home_weekend   alphaTH      etaTH   alphaEP  alphaI1D   alphaHD  alphaI1H
## 1     0.03887666 0.7898453 0.02628018 -2.787419 -19.90561 -2.413713 -1.978009
##          eta repeats  output
## 1 0.01982861      10 Ens0000
```

row 2 is input vector 2, etc:

```
inputs[2,]
```

```
##         R0       TE       TP      TI1        TI2       nuA       ns
## 2 2.256901 1.001359 3.347003 4.017073 0.04740412 0.1350559 22.10835
##   p_home_weekend  alphaTH      etaTH   alphaEP  alphaI1D   alphaHD  alphaI1H
## 2      0.6350796 0.534793 0.03323681 -3.151488 -19.55221 -3.524098 -2.000956
##          eta repeats  output
## 2 0.02323521      10 Ens0001
```

The 1st 15 columns are the different inputs, e.g. R0, … The 'repeats' column is how many times the simulator was run at this particular set of inputs. This is because the model is stochastic - it may be useful to run it multiple times in the same place. Output is a tag so that we can match up this input vector with the COVID simulator output.

The full output database is extremely large. I can share this at a later date, but it makes sense to start off with a smaller version. In this version, I've already aggregated to LAD (Local Authority District) and by age. This matrix still has 1.6 million rows:

```
dim(outputs)
```

```
## [1] 1661100      10
```

```
head(outputs)
```

```
##    output replicate   LAD19CD cumH cumHD cumCD deaths    LAD19NM    region week
## 1 Ens0000         1 E06000001   94    11     0     11 Hartlepool E12000001   12
## 2 Ens0001         1 E06000001    3     0     0      0 Hartlepool E12000001   12
## 3 Ens0002         1 E06000001 2689   197     0    197 Hartlepool E12000001   12
## 4 Ens0003         1 E06000001    3     0     0      0 Hartlepool E12000001   12
## 5 Ens0004         1 E06000001    2     0     0      0 Hartlepool E12000001   12
## 6 Ens0005         1 E06000001   25     0    18     18 Hartlepool E12000001   12
```

Each row gives the number of hospitalisations (cumH), hospital deaths (cumHD), community deaths (cumCD), and deaths, by:

- `output` - which row of the input matrix did we use to run the COVID simulator?
- `replicate` - sometimes we run the simulator more than once
- `LAD19CD`, `LAD19NM` - name and code for the LAD
- `region` - code for what region this LAD is in
- `week` - which week is it? Between 6 and 12 (weeks since start of simulation)

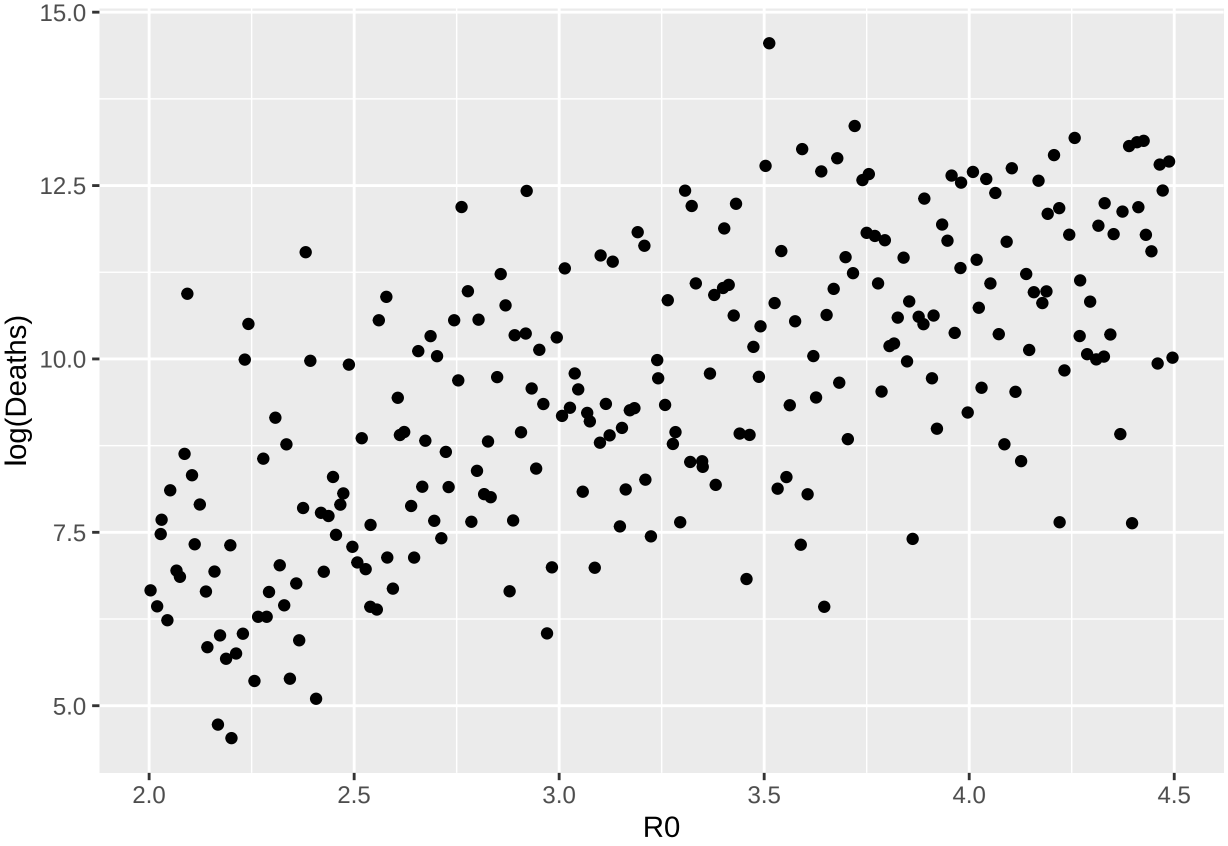
So to see the output generated by a SINGLE simulation run:

```
single_sim <- subset(outputs, output == 'Ens0000' & replicate == 1)
dim(single_sim)
```

```
## [1] 2373   10
```

Here's a very crude initial plot, of R0 vs deaths in week 12:

```
tmp <- subset(outputs, replicate == 1 & week == 12)
totals <- aggregate(deaths ~ output + week, data = tmp, FUN = sum)
ggplot(data.frame(R0 = inputs$R0, Deaths = totals$deaths), aes(x = R0, y = log(Deaths))) +
  geom_point()
```



So unsurprisingly, there's a clear signal (higher R0 = more deaths), but it's quite noisy - there's dependence on other input parameters.

This is also only 1 summary of the output - each of these dots is representing a sum across all LADs at this timepoint, so we could plot, explore and model spatial and/or temporal aspects of the simulations.

Land surface modelling

Data coming soon - new runs underway.

JULES is a land surface model that can be used to simulate the interaction between climate/CO2 and vegetation. This data comes from a project that is interested in understanding where trees should be planted in the UK, in order to best reduce the amount of CO2, under different climate scenarios.

Air quality

As a new alternative, this dataset contains 31 simulations of the UK Air Quality model. The inputs are emissions across different sectors; the outputs are concentrations of different pollutants at a number of sites across the UK, over time.

There are 161 spatial locations across the UK, and concentrations of NO2, SO2, O3, PM2.5 and PM10 are reported hourly for 745 hours.

The inputs for this model - as with the others, each row indicates the inputs used for a single simulator run:

```
##          S1        S2 S3 S4 S5 S6 S7 S8 S9 S10     ID
## 1 1.0000000 1.0000000  1  1  1  1  1  1  1   1   CTRL
## 2 0.3333333 1.0000000  1  1  1  1  1  1  1   1 S1_F03
## 3 0.6666667 1.0000000  1  1  1  1  1  1  1   1 S1_F06
## 4 1.3333333 1.0000000  1  1  1  1  1  1  1   1 S1_F13
## 5 1.0000000 0.3333333  1  1  1  1  1  1  1   1 S2_F03
## 6 1.0000000 0.6666667  1  1  1  1  1  1  1   1 S2_F06
```

S1, . . . , S10 refer to emissions from different sectors. 1 is the baseline value, and these are scaled up or down by changing this input. The design here is a bit more systematic than the Covid one - here each input is varied individually. The `ID` refers to tags in the filenames.

Similarly to other datasets, the simulator output is provided in NetCDF format, with a single file for each of the 31 simulations. Reading in the `aq_example.nc`:

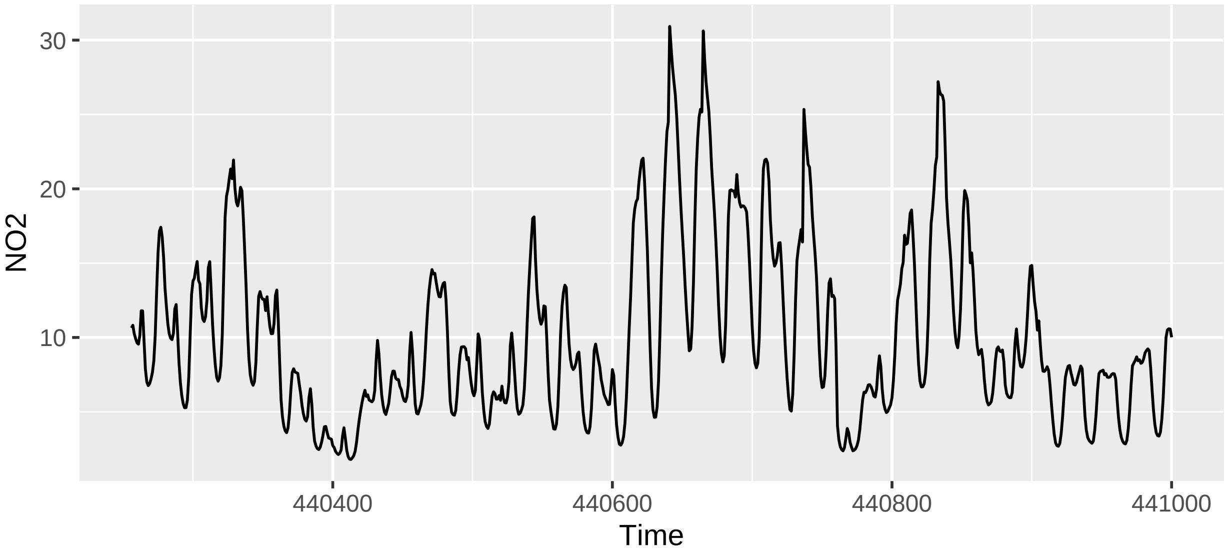
```
example <- nc_open('~/Dropbox/UQ_projects/Data/air_quality/aq_example.nc')
lon <- ncvar_get(example, varid = 'longitude')
lat <- ncvar_get(example, varid = 'latitude')
time <- ncvar_get(example, varid = 'time')
example_NO2 <- ncvar_get(example, varid = 'mass_concentration_of_nitrogen_dioxide_in_air')
dim(example_NO2)
```

```
## [1] 745 161
```

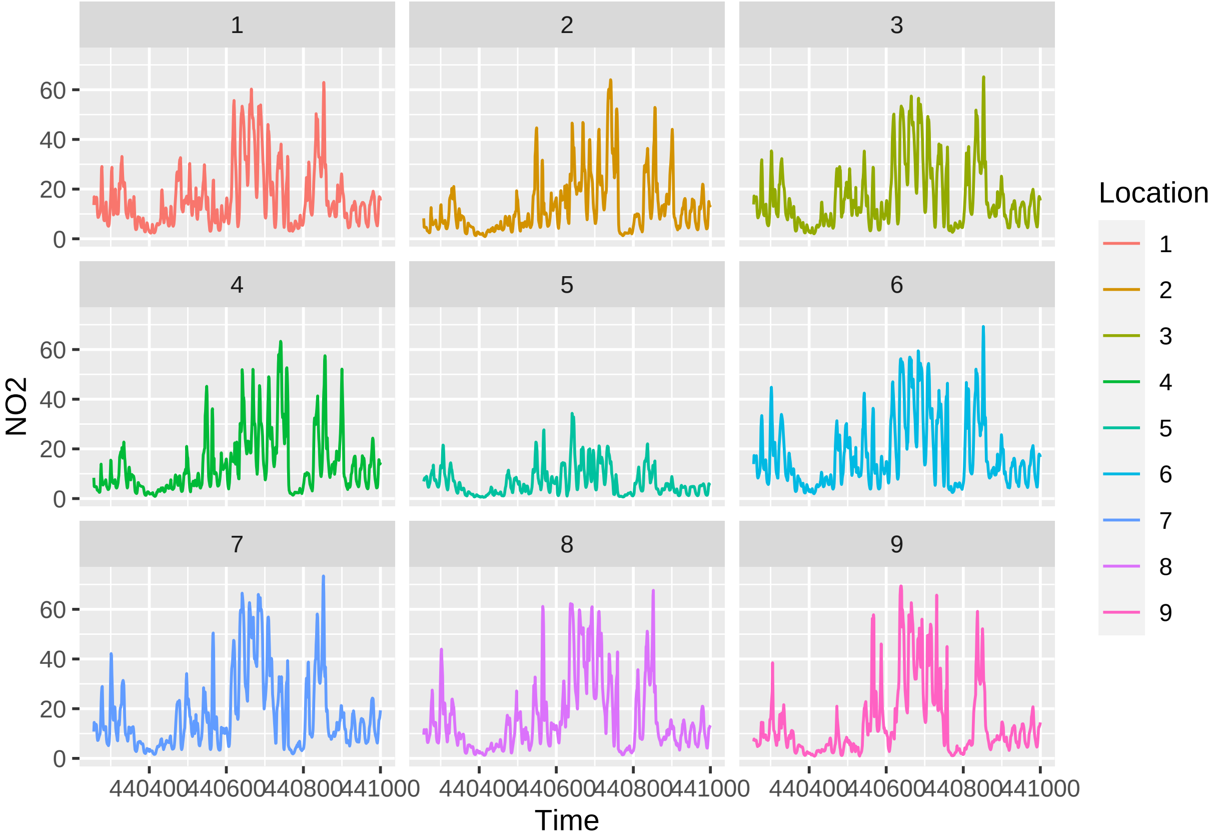
So this is the NO2 field for this particular simulation. We could plot different aspects of this, e.g. time series in a specific spatial location, time series averaged across all spatial locations, diurnal cycle (24 hour cycle, averaged across the 31 days), the mean spatially, etc. And like with the dispersion examples, this is just a single simulation! We want to investigate the effect of the input parameters, here representing sector emissions, and see whether we can find something consistent with the observations (or, more likely - identify where we **can't** do this - we might be able to do well for certain regions, or certain pollutants, but unlikely to be able to capture the observations everywhere - but this is useful information, and definitely an interesting project result).

Other interesting aspects: if we only consider e.g. the 1st week of the simulation, how well can we emulate the model? How different are calibration results as we change the amount of training data? (this is useful as want to minimise the length of expensive simulations - if we don't learn anything more after a certain day then we shouldn't bother with longer simulations).

Here's a very simple plot of the full NO2 time series **for this particular simulation**, averaged across all locations:



Or considering a few different locations separately:

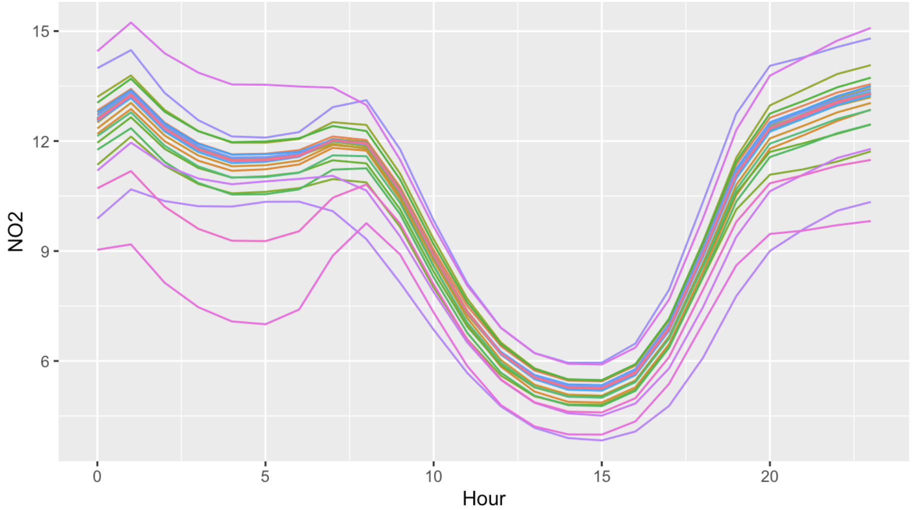


Important: time series modelling isn't usually helpful here - this might look like a time series modelling problem, but it's not:

- we don't have a single time series, e.g. of real-world observations, that we want to model
- instead we have 31 different simulations of the real-world, and we want to model **these** (and explore the dependence on the inputs to the simulator, rather than the dependence on time)
- so any model we fit needs to have dependence on the simulator inputs

- and our simulator can produce infinitely-many versions of the real-world if we vary the inputs
- we're trying to **calibrate** with respect to the simulator inputs, rather than forecasting ahead, and to do so we need to be able to model dependence across (or interpolate between) the inputs
- there are also non-linearities that may be difficult to capture - because the simulator itself is extremely complex and simulating interactions and reactions between different pollutants in the atmosphere - the true simulator will definitely be better at this than a simple time series model!

To highlight further that we want to capture the dependence on the inputs, not on time: the plot below takes all 31 simulations, averages NO2 over all spatial locations, and averages to a 24 hour cycle (diurnal cycle):



There's generally the same sort of pattern, but there's clear dependence on the inputs, that shifts the strength of this cycle up or down. There's also different patterns sometimes - so there's probably more complex structure than just a linear dependence on the inputs.

Ideally, given a new set of inputs, we want to predict **the whole time series**. In practice, we might just predict the average value of something, or at a particular location, or at a particular timepoint.

Others

There's lot of climate data out there, e.g. future warming scenarios, which could be explored, linked into other things (e.g. increased heat in future = more heat wave risk = greater effect on health?)

CEDA is a repository of all sorts of climate data: https://catalogue.ceda.ac.uk/?q=ukcp18&sort_by=