Bocconi University, 20236 - Time Series Analysis

Final project

Get started with your final project.

As you know, the exam of the course 20236 includes a "final project" on real data analysis with R.

We propose a "running dataset" for you, introduced below. You are however free to propose another problem and dataset of your choice, if interested. In that case, please first discuss your idea with us.

We encourage you to start working on your final project starting from now.

You will receive a brief feedback by your tutors, Michele and Filippo; and you may modify/improve your analysis and presentation before submitting the final project.

The first steps will be

- description of the problem and questions of interest ("motivation")
- description of the data, also providing the source ("collect info")
- exploratory analysis (describe and visualize with plots, summarize information)
- Address the first question. (Here you will likely have a modeling step; and estimation and prediction; and model evaluation).

1. Description of the problem and questions of interest

Air pollution is a serious issue that severely impacts human health. In particular, the link between respiratory diseases and the presence of particulate matter has been extensively studied (see Dominici et al. (2006) *Fine particulate air pollution and hospital admission for cardiovascular and respiratory diseases.* JAMA, 1127-1134); levels of $PM_{2.5}$ and PM_{10} (particulate matter of diameter 2.5 and 10 micrometer or less, respectively) may be associated to more severe Covid19 outcomes (Wu et al. (2020) *Air pollution and COVID-19 mortality in the United States: Strengths and limitations of an ecological regression analysis.* Sci. Adv.).

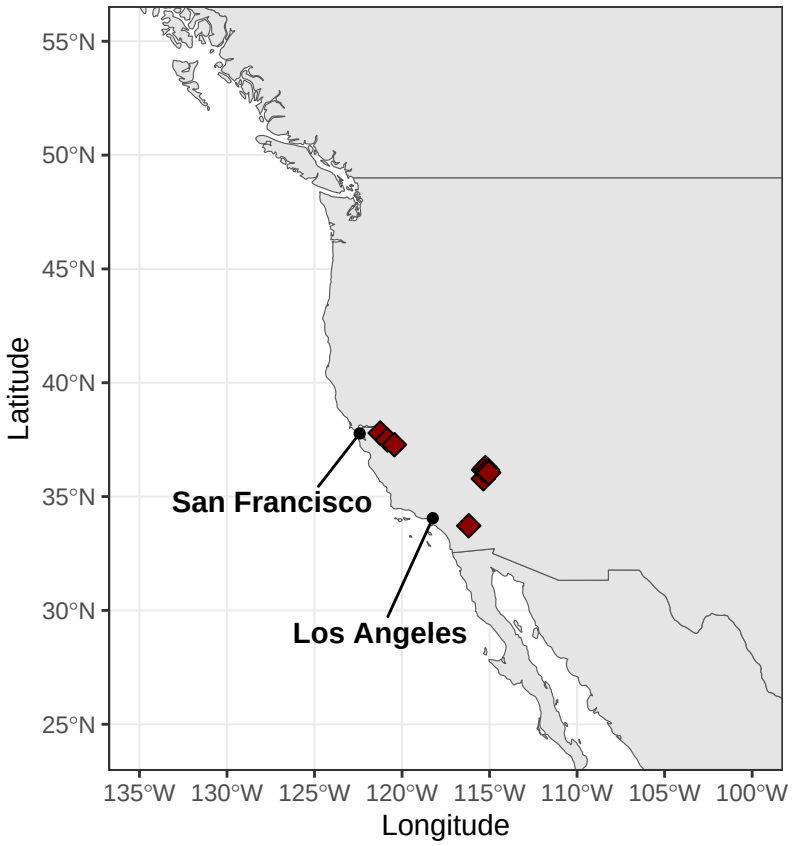
With the goal of modeling the dynamics of air pollution, we consider hourly air quality data from the U.S. Environmental Protection Agency (EPA). We consider data from 10 stations located along the U.S. West Coast over a period that covers summer 2020, including the 2020 wildfire season. The raw data can be downloaded from the EPA website. Notice that each station may yield missing data, possibly because failing a validation step performed by EPA.

A thorough statistical analysis may be particularly useful to politicians and decision makers, in order to suggest the best behaviour to the citizens. Interesting questions are

- We might want to identify different levels of pollution and instability, that may require different interventions from the decision makers. How could we identify and estimate these different levels, from the data? If we have a high level of pollution at a certain time, can we predict that it will remain such in the next hour? Can we quantify the probability to see a significant decrease in the next few hours?
- The dataset refers to the summer 2020; in fact, the original data were streaming in, hour-by-hour. Can we provide online estimation and prediction with streaming data? In particular, can we provide uncertainty quantification for such predictions? Hourly data can be very noisy and irregular: should we look at another time-scale, for example at daily (or half-daily) averages? What changes?
- Can we model the different stations jointly? In other words, can we incorporate the spatial dimension in our analysis? Do we get further insights from this analysis?

2. Data description

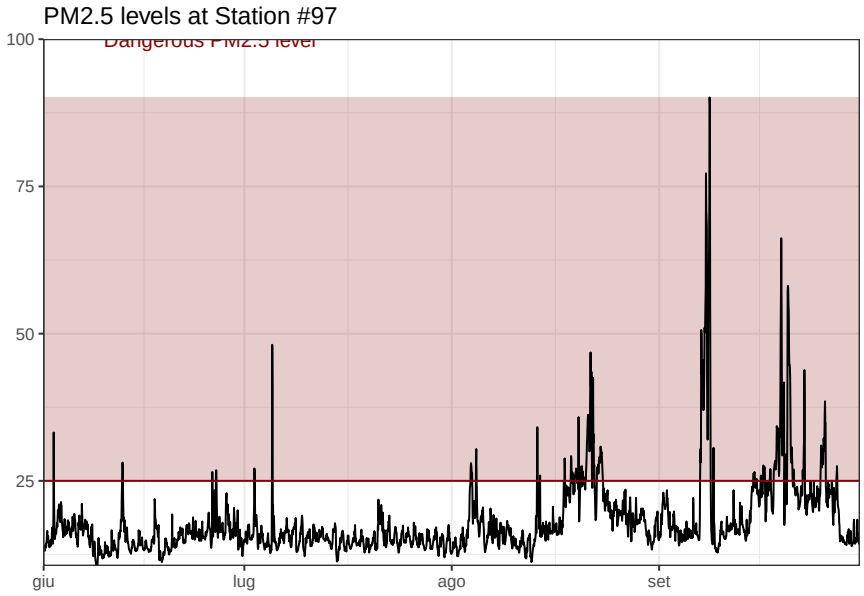
To study the problem and address the relevant questions we need to collect appropriate data. For our purposes, as already said, we consider space-temporal data from 10 stations in California. They are characterized by having no missing values (NA) for $PM_{2.5}$, and by a long period of measurements, from June to September 2020. In particular, the first observation reported is taken at 00.00 GMT (that means 5 pm in San Francisco). The next map shows the positions of the stations: they lie approximately between San Francisco and Los Angeles.



In particular, the dataset includes:

- `Longitude` and `Latitude`: the spatial coordinates of the EPA station
- `datetime`: the timestamp (GMT time zone)
- `pm25`: particulate matter of size 2.5 micrograms per cubic meter or less, over the minimum recorded in the data.
- `temp`: air temperature in Celsius.
- `wind`: wind speed in knots/second.
- `station_id`: station identifier within this dataset.

As an example, consider station 97 between San Francisco and Los Angeles. Notice that the suggested limit of $PM_{2.5}$ is given by 25 micrograms per cubic meter (average over 24 hours).



The majority of the measurements show values smaller than the prescribed limit. However, the dynamic of the particulate matter shows some high peaks, probably given by the outburst of fires; the latter can be caused by high temperatures and severely exacerbated by wind. Therefore, the three time series are likely correlated.

First part of the assignment

Thus, to get started, you should choose a station (to be chosen e.g. among the ones with id 47, 55, 92, 95, 97 and 103) and

- Describe suitably the time series, with appropriate plots and comments.
- Try a Gaussian HMM with a suitable number of states (motivate/comment your choice). Can you answer the first question of interest?

Make sure to tune the analysis to the specific station considered.

Second part

The goal is to address the second and third set of questions presented previously. In particular, we want to introduce the spatial dimension in our model.

Here is a fairly simple but already rather interesting model for these *spatio-temporal* data.

Let $(Y_{j,t}, \theta_{j,t})$ denote, respectively, the observed measurement and the signal in station j at time t. For a single station, let's consider a simple random walk plus noise model

$$\begin{cases} Y_{j,t} = \theta_{j,t} + v_{j,t} \\ \theta_{j,t} = \theta_{j,t-1} + w_{j,t}, \end{cases}$$

with the usual assumptions on $\theta_{j,0}$ and the errors sequences.

Yet, in order to take the spatial dependence into account, we need a multivariate model for the m-dimensional $Y_t = (Y_{j_1,t}, \dots, Y_{j_m,t})'$ - for us, the PM_{25} observeded at stations $j_1, \dots, j_m$ (take $m = 3$ or 4). We may write a DLM

$$\begin{cases} Y_t = F\theta_t + v_t & v_t \overset{indep}{\sim} N_m(\mathbf{0}, V) \\ \theta_t = G\theta_t + w_t, & w_t \overset{indep}{\sim} N_p(\mathbf{0}, W) \end{cases}$$

where each $(Y_{j,t})$ is described as a random walk plus noise as above (*we leave to you to write down what are F, G, θ_t*). A common assumption is that the measurement errors $v_{j,t}$ are independent across locations j, with location-specific variances; while spatial dependence is modeled through the evolution errors. That is, in the multivariate DLM, assume that V is a diagonal matrix

$$V = \begin{bmatrix} \sigma^2_{v,1} & 0 & 0 & 0 \\ 0 & \sigma^2_{v,2} & 0 & 0 \\ 0 & 0 & \sigma^2_{v,3} & 0 \\ 0 & 0 & 0 & \sigma^2_{v,4} \end{bmatrix}$$

while the evolution errors $w_t = (w_{j_1,t}, \ldots, w_{j_m,t})$ are spatially correlated; that is, the covariance matrix W is not diagonal and expresses the intuitive idea that the larger the distance between two stations, say j and k, the smaller is the covariance $W[j,k] = Cov(w_{j.t}, w_{k,t})$.

The choice of the covariance function for spatial processes is a broad topic and an important element of the model specification. Here we consider a popular choice, namely an exponential covariance function such that

$$W[i,k] = Cov(w_{j.t}, w_{k,t}) = \sigma^2 \exp(-\phi D[j,k]), \quad j,k = 1, \ldots, m,$$

where $\sigma^2 > 0$; $\phi > 0$ is a *decay parameter*; and $D[j,k]$ is the distance between stations j and k (see below about computing the $m \times m$ matrix D of the pairwise distances among the stations). Note that this specification assumes that $V(w_{j,t}) = \sigma^2$, constant across locations, and expresses spatial dependence: the $Cov(w_{j.t}, w_{k,t})$ gets smaller, the more far apart are the stations j and k.

Our suggestion is that you try this model for the $PM_{2.5}$, after some elaboration of the data (suggested to simplify your analysis):

- Choose 3 or 4 stations (at least two should be chosen among the ones with id 47, 55, 92, 95, 97 and 103) and extract the associated time series;
- Use a log scale (that is, take the log), to smooth sharp picks.
- Moroever, to simplify your analysis (smoothing the irregularities of hourly data), work with a coarser scale, e.g. by taking 12-hours averages.

The resulting time series will be the object of the analysis.

- Look at the data: describe the time series of $PM_{2.5}$ through appropriate plots. Do the plots suggest spatial dependence?
- Model specification: you may use the joint spatio-temporal model described above. [To compute the distance between two stations, you can use the Euclidean distance between the corresponding pairs of longitudes and latitudes (or use a more sophisticated approach, e.g. converting them to meters).]
- Estimate the unknown parameters by maximum likelihood – report accurately the results.
- Provide one-step-ahead predictions.
- Comment on the values of the estimated parameters and the resulting predictions.
- Final comments. Are you satisfied with the model? Are you making reasonable or restrictive assumptions through the model? (*Assuming a model means including information, and the results do rely on that information. It may be insightful to compare the assumptions of the (univariate) HMM used in the first part, and the assumptions of the (univariate) model used here.*).

Remember that the final project is also a useful exercise of *presentation*.

Below you find suggestions on how your analysis should be presented (they were posted, and are still available, on BBoard).

Submission

- Single .zip file with report and code to reproduce
- Report in pdf (if from rmarkdown: Knit to PDF. do not export HTML and then print)
- Code in .R or .rmd
- Name of zip file = group name

Length

- The PDF file must be no longer than 8 pages.

First page includes

- Group name
- Names of group components
- Scientific question you attempt to answer, and how (briefly)

Format (specific to final project but not for the assignments). Remember: you are supposed to send your code so the report should not include any!

- NO R console output: use tables
- NO R messages
- NO R code anywhere ever
- NO code chunks
- NO mention of the functions you use, and no explanation of your code
- Can the report be read 100% the same way if the code was not written in R? If yes, then good; if not, then make it independent of the code. The analyses and your interpretations are important, not the specifics of your code. Good code will lead to more elegant analyses & plots & overall presentation
- NO screenshots

Contents

- All models are written in formulas
- Notation is consistent
- Estimates for all unknowns are reported in tables/plots or discussed in the text and interpreted
- Uncertainty quantification of estimates and brief interpretation
- Model comparisons are meaningful

Plots and figures

- All plots have short description/title/caption and are numbered
- All figures numbered sequentially
- All figures are mentioned in text, in the order in which they appear
- All plots have meaningful axis titles (if not redundant e.g. in the title)
- All plots are well-positioned in the page (centered)
- All text in the plots is readable without zooming in
- No text is too big in the plot
- Plots are not "warped"
- No plot is pixelated or blurry or with jpeg artifacts
- All plots are useful for the purpose of answering the research question
- All plots are explained and interpreted (not just described passively)

General

- Spacing is used efficiently: no excessive white spaces
- Borders are normal, line spacing is standard, no other weirdness to fit everything within the page limit

- English: spelling mistakes? Too verbose? Concise enough? We're not the British Council but you don't want to be sloppy.
- Report does not look hastily made or sloppy
- Report looks professional
- Text is concise and to the point

Code – we will randomly pick some groups for a code check. Or, we may check the code when figures or values look funny (as it happens)

- Submitted code can be compiled/run without error generating all figures and tables in the report, with the same numbers
- Code is easily readable and it is possible for anybody to understand what is going on
- Variables are named to improve readability (i.e. avoid calling things "a1" "x9534", "asdfa", but rather use names such as "user_speed", "daily_price", "log_returns".
- The code would work with minor modification on different data