Dídac Roda Pitarg

FROM EMISSIONS TO ADMISSIONS: EXPLORING THE AIR QUALITY OF CAMP DE TARRAGONA AND ITS IMPACT ON PEDIATRIC RESPIRATORY HEALTH USING MACHINE LEARNING

FINAL DEGREE PROJECT

Codirected by Dr. Maria Vinaixa and Dr. Noelia Ramírez

Bachelor's Degree in Biomedical Engineering

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

En primer lloc, m'agradaria donar les gràcies a la meva família, pel seu recolzament, la seva compressió i fe en mi al llarg d'aquests durs mesos.

Agrair a la meva tutora, la Maria, per la seva ajuda al llarg de tot el treball.

També m'agradaria expressar el meu més sincer agraïment a la Noelia per brindar-me l'oportunitat de treballar com a enginyer en el projecte OnBREATHE. Aquesta experiència ha estat i està sent increïblement enriquidora.

Gràcies al Joaquín i a l'Amalui per ajudar-me a aconseguir les dades clíniques.

Finalment, un reconeixement especial als meus companys del grup de Toxicologia i Metabolòmica Ambiental, la Mahsa, la Camilla, el Pau i l'Alberto per recolzar-me i animar-me al llarg d'aquests mesos.

Abstract

Children are particularly vulnerable to the health consequences of air pollution. This study explores the potential link between air quality and pediatric respiratory health in the Camp de Tarragona area, a region with a major petrochemical industry.

We analyzed air quality data (2013-2023) from the Air Pollution Surveillance and Prediction Network (XVPCA) alongside pediatric hospital admission data for respiratory illnesses from Sant Joan de Reus Hospital (HUSJR). Spatial variations in pollutant concentrations emerged, with higher levels in urban areas. A notable decrease in pollutants coincided with the 2020 COVID-19 lockdown. Analysis of admissions identified acute nasopharyngitis, bronchitis, and bronchiolitis as the most frequent diseases. Admissions peaked in children under one year old.

Spearman's rank correlation analysis suggested weak positive correlations between PM2.5, PM10, and NO_x concentrations with hospital admissions. The strongest correlation was observed between weekly NO_x exposure and total admissions after log transformation. Machine learning models using Support Vector Regression (SVR) and Partial Least Squares (PLS) regression achieved limited predictive power (accuracy of 0.18 and 0.25, respectively) in estimating admissions based on air pollutant concentrations.

This study suggests a potential association between air pollution and pediatric respiratory health in the Camp de Tarragona region. Further research is needed to solidify this link and inform public health interventions. Limitations include reliance on data from one hospital and the use of monitoring station data, not individual exposure levels. Future research will incorporate broader pediatric data, individual exposure assessment, weather conditions, and a wider range of pollutants, potentially using more advanced machine learning models.

Resumen

Los niños y niñas son especialmente vulnerables a las consecuencias para la salud de la contaminación atmosférica. Este estudio explora la posible relación entre la calidad del aire y la salud respiratoria pediátrica en la zona del Camp de Tarragona, una región con una importante industria petroquímica.

Se analizaron datos de calidad del aire (2013-2023) de la Red de Vigilancia y Predicción de la Contaminación Atmosférica (XVPCA) junto con datos de ingresos hospitalarios pediátricos por enfermedades respiratorias del Hospital Sant Joan de Reus (HUSJR). Se observaron variaciones espaciales en las concentraciones de contaminantes, con niveles más altos en las zonas urbanas. Un descenso notable de los contaminantes coincide con el confinamiento por la COVID-19 en 2020. El análisis de los ingresos identificó la nasofaringitis aguda, la bronquitis y la bronquiolitis como las enfermedades más frecuentes. Los ingresos alcanzaron su máximo en niños menores de un año.

El análisis de correlación de rangos de Spearman sugirió correlaciones positivas débiles entre las concentraciones de PM2,5, PM10 y NO_x con los ingresos hospitalarios. La correlación más fuerte se observó entre la exposición semanal a NO_x y el total de ingresos tras la transformación logarítmica. Los modelos de aprendizaje automático mediante regresión por vectores de apoyo (SVR) y regresión por mínimos cuadrados parciales (PLS) alcanzaron un poder predictivo limitado (precisión de 0,18 y 0,25, respectivamente) en la estimación de ingresos basada en las concentraciones de contaminantes atmosféricos.

Este estudio sugiere una posible asociación entre la contaminación atmosférica y la salud respiratoria pediátrica en el Camp de Tarragona. Se necesitan más investigaciones para consolidar este vínculo y fundamentar las intervenciones de salud pública. Las limitaciones incluyen la dependencia de los datos de un solo hospital y el uso de datos de estaciones de monitorización, no de niveles de exposición individuales. La investigación futura incorporará datos pediátricos más amplios, evaluación de la exposición individual, condiciones meteorológicas y una gama más amplia de contaminantes, utilizando potencialmente modelos de aprendizaje automático más avanzados.

Resum

Els nens i nenes són especialment vulnerables a les conseqüències per a la salut de la contaminació atmosfèrica. Aquest estudi explora la possible relació entre la qualitat de l'aire i la salut respiratòria pediàtrica en la zona del Camp de Tarragona, una regió amb una important indústria petroquímica.

Es van analitzar dades de qualitat de l'aire (2013-2023) de la Xarxa de Vigilància i Predicció de la Contaminació Atmosfèrica (XVPCA) juntament amb dades d'ingressos hospitalaris pediàtrics per malalties respiratòries de l'Hospital Sant Joan de Reus (HUSJR). Es van observar variacions espacials en les concentracions de contaminants, amb nivells més alts en les zones urbanes. Un descens notable dels contaminants coincideix amb el confinament per la COVID-19 en 2020. L'anàlisi dels ingressos va identificar la nasofaringitis aguda, la bronquitis i la bronquiolitis com les malalties més freqüents. Els ingressos van aconseguir el seu màxim en nens menors d'un any.

L'anàlisi de correlació de rangs de Spearman va suggerir correlacions positives febles entre les concentracions de PM2,5, PM10 i NO_x amb els ingressos hospitalaris. La correlació més forta es va observar entre l'exposició setmanal a NO_x i el total d'ingressos després de la transformació logarítmica. Els models d'aprenentatge automàtic mitjançant regressió per vectors de suport (SVR) i regressió per mínims quadrats parcials (PLS) van aconseguir un poder predictiu limitat (precisió de 0,18 i 0,25, respectivament) en l'estimació d'ingressos basada en les concentracions de contaminants atmosfèrics.

Aquest estudi suggereix una possible associació entre la contaminació atmosfèrica i la salut respiratòria pediàtrica al Camp de Tarragona. Es necessiten més recerques per a consolidar aquest vincle i fonamentar les intervencions de salut pública. Les limitacions inclouen la dependència de les dades d'un sol hospital i l'ús de dades d'estacions de monitoratge, no de nivells d'exposició individuals. La recerca futura incorporarà dades pediàtriques més àmplies, avaluació de l'exposició individual, condicions meteorològiques i una gamma més àmplia de contaminants, utilitzant potencialment models d'aprenentatge automàtic més avançats.

Contents

List of Abbreviations

API	Application Programming Interface
CEIm	Ethics Research Committee
DCC	Digital Curation Centre
DMP	Data Management Plan
HUJ23	Hospital Universitari Joan XXIII
HUSJR	Hospital Universitari Sant Joan de Reus
ICD	International Classification of Diseases
ICQA	Catalan Air Quality Index
IDESCAT	Catalan Statistics Institute
IoT	Internet of Things
ML	Machine Learning
NMVOC	Non-Methane Volatile Organic Compound
NOx	Nitrogen Oxides
PLSR	Partial Least Squares Regression
PM	Particulate Matter
SDG	Sustainable Development Goal
SOx	Sulfur Oxides
SVR	Support Vector Regression
VOC	Volatile Organic Compound
XVPCA	Air Pollution Surveillance and Prediction Network

List of Figures

List of Tables

List of Codes

1 Introduction

Children are particularly vulnerable to the adverse effects derived from air pollution. Their developing respiratory systems are particularly susceptible to the quality of air to levels of pollution they are exposed. Exposure to toxicants during this critical period can lead to a lifetime of health problems [1]. This research project investigates the link between air pollution air pollution and pediatric respiratory health in "Camp de Tarragona", an area in Catalonia sourounded by the largest petrochemical complex in South Europe.

The "Camp de Tarragona" area presents an important concentration of petrochemical industries and thus, a variety of air pollutants, including volatile organic compounds (VOCs), particulate matter (PM), and nitrogen oxides (NO_x) among others are continuously monitored to control air quality. These pollutants are linked to a range of respiratory issues in children, including asthma, bronchitis, and pneumonia [2]. This study seeks to shed light on the relationship between air quality and pedriatric respiratory health. We will use different machine learning (ML) approaches to analyze the relationship between air pollution data and those pediatric hospital admissions with diagnosis for respiratory illnesses. This research can contribute to a growing body of knowledge on the health impacts of air quality. The findings can inform public health policy decisions aimed at reducing emissions, improving air quality, and protecting the health of children in the "Camp de Tarragona" and similar communities worldwide.

The following chapters will delve deeper into the existing literature on air pollution and pediatric respiratory health, detail the methodology employed in this study, present the results of the analysis, and discuss the implications of our findings for public health and future research directions.

1.1 Air quality

Air pollution poses a persistent and severe environmental threat worldwide, directly impacting human health, particularly among vulnerable groups like children. In urban and industrial areas, levels of pollutants often exceed the safety limits established by environmental protection agencies [3]. This situation is particularly concerning in the "Camp de Tarragona", an industrially active area with a significant population density.

In adult population, epidemiological studies have demonstrated a link between air pollution and various adverse health conditions (mainly respiratory and cardiovascular) [4]. However, pedriatric population are much less studied.

The "Joan XXIII" and "Sant Joan de Reus" hospitals in Tarragona possess comprehensive data on pediatric health issues, including respiratory, cardiovascular, and other related illnesses. Additionally, the Air Pollution Surveillance and Prediction Network (XVPCA) systematically records pollutant levels across its monitoring stations in the "Camp de Tarragona".

This study aims to address a gap in the scientific literature regarding pediatric health and air pollution in the "Camp de Tarragona" region. By analyzing data from hospitals

and the Surveillance Network, we seek to establish significant correlations between air pollutant levels and the incidence of childhood illnesses. This information will not only improve our understanding of pollution's impact on this vulnerable group but also guide future prevention and intervention policies to enhance pediatric health in the region.

1.2 The *OnBREATHE* project

The project this work contributes to is called *OnBREATHE*: "Personal air quality monitoring and data digitalization to track chronic respiratory diseases", funded by ISCIII (PMPTA22/0028). This ambitious proposal brings together a team of eighteen researchers with diverse expertise. They hail from four distinct research groups: the Pediatric, Nutrition and Development Research Unit (PEDINUR) at the Institute of Health Research Pere Virgili (IISPV), the Microsystems Nanotechnologies for Chemical Analysis (MINOS) and Chromatography and Environmental Applications (CROMA) groups from the University of Rovira i Virgili, and a collaborating group from the Biomedical Research Center for Respiratory Diseases Network (CIBERES) at "La Princesa" hospital in Madrid. This collaboration across various fields, from environmental health to pediatric pneumology, underscores the project's multifaceted approach to tackling chronic respiratory diseases with the collaboration of Onlean company, in charge of data digitalization.

OnBREATHE's core mission is to create and verify a groundbreaking wearable digital tool (Figure 1). This tool will track both air quality and health markers in adult and pediatric patients with chronic respiratory illnesses, allowing for personalized monitoring of their exposure to organic pollutants. This solution has two specific objectives:

1. **Develop a monitoring device** that will track a patient's exposure to air pollutants with two functionalities:
   1.1. **Offline characterization** of captured volatile organic compounds (VOCs) a patient is exposted to over a period of time.
   1.2. **Continuous monitoring** with a built-in microsensor that will track real-time air quality by measuring temperature, humidity, pressure, and VOCs concentration.
2. **Create a smartphone app and a software platform** that will record and combine exposure data with the patient's health information and disease progression.
   2.1. **Data collection** about their health status, activity levels and exposure and allowing to download the continuous microsensor data to patients.
   2.2. **Data analysis** and processing by the researchers of the collected parameters stored in the software platform.

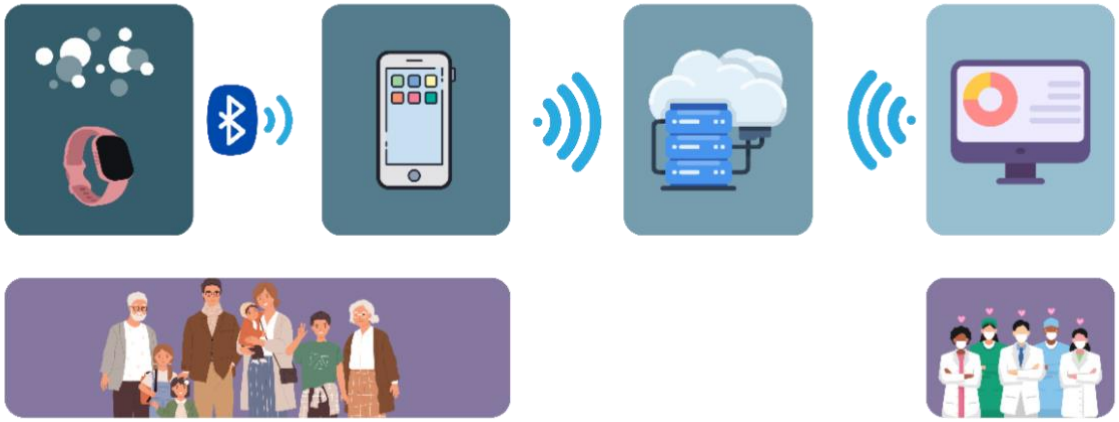


Figure 1. Graphical abstract of the OnBREATHE project.

This work focuses on the link between air pollution and pediatric respiratory health in the "Camp de Tarragona", directly complements the ambitious goals of the *OnBREATHE* project. While *OnBREATHE* develops a valuable tool for personalized air quality monitoring, understanding the specific health effects of pollutants on children in a heavily industrialized region like ours is crucial. The findings can inform the development of the *OnBREATHE* platform by pinpointing the most relevant air pollutants to track for pediatric patients residing in similar environments. This targeted approach can enhance the effectiveness of the *OnBREATHE* tool for this vulnerable population. Additionally, this research can contribute to the project's data analysis phase by providing a baseline understanding of how air pollution levels correlate with the incidence of respiratory illnesses in children. This deeper knowledge will strengthen the interpretation of data collected by the *OnBREATHE* wearable device, ultimately leading to more personalized treatment plans and improved health outcomes for children with chronic respiratory conditions.

1.3 Sustainable Development Goals

In a historic move in 2015, the United Nations member states adopted the 2030 Agenda for a new era of sustainability. This comprehensive strategy delineates a route to attain financial prosperity while safeguarding the environment and its inhabitants.

With climate change a growing threat, the seventeen Sustainable Development Goals (SDGs) offer a clear path forward to ensure a sustainable future for generations to come.

This project contributes to some of the SDGs (Figure 2).

Figure 2. SDGs related to the project.

Goal 3: Good health and well-being, prioritizes ensuring healthy lives and well-being for all. This study directly aligns with SDG 3 by investigating how air pollution impacts pediatric respiratory health. Children are a particularly vulnerable population, and understanding these health effects can inform interventions to protect their well-being.

Goal 11: Sustainable cities and communities, emphasizes creating sustainable and healthy cities. This study examines the health effects of industrial facilities in the "Camp de Tarragona", contributing valuable insights for establishing sustainable urban environments, especially those with heavy industry. By understanding the health impacts, policymakers can create regulations and urban planning strategies that promote both economic development and the well-being of citizens.

Goal 13: Climate action. Understanding the health consequences of air pollutants can inform policy decisions that promote cleaner energy sources, ultimately contributing to climate action efforts.

By aligning with these specific SDGs, the project takes a multifaceted approach to building a more sustainable future.

1.4 Motivation

My journey towards this research project began in my hometown of Vila-seca, nestled within the "Camp de Tarragona" industrial complex. Raised in the shadow of petrochemical factories, the potential impact of their emissions on public health was a constant concern. This concern morphed into a personal mission as I witnessed loved ones struggle with respiratory issues.

Biomedical engineering offered a path to understand the biological mechanisms at play in respiratory illnesses and provided the tools to collect and analyze the vast amount of environmental and health data that could hold the key to unlocking solutions. My fascination with data analysis boosted my desire to bridge the gap between these disciplines.

The confluence of these interests led me to explore the field of environmental health, where sophisticated data analysis techniques can be used to investigate the relationship between environmental factors and health outcomes. This research project represents the culmination of these converging interests, specifically focusing on the

detrimental effects of air pollution on pediatric respiratory health in the "Camp de Tarragona".

The proximity of the research to my personal experiences provides a powerful motivator. By delving into this topic, I not only contribute to the advancement of scientific knowledge but also aim to find solutions that directly impact the health of my community, especially children who are particularly vulnerable to environmental pollution. The potential to translate complex data analysis into tangible solutions that improve the lives of those around me is a deeply motivating force driving this research forward.

1.5 Hypothesis and goals

Understanding the potential health impacts of air pollution on children residing near industrial areas is crucial for developing effective prevention and intervention strategies. To achieve this, the following hypotheses are proposed:

- **Hypothesis A**: there is a significant correlation between the levels of air pollutants measured at different stations in the "Camp de Tarragona" and the cases of pediatric respiratory health problems registered by the "Joan XXIII de Tarragona" and "Sant Joan de Reus" hospitals.
- **Hypothesis B**: air pollutants, especially PM2.5 and PM10, have a positive correlation with respiratory diseases in pediatric respiratory health.
- **Hypothesis C**: monitoring stations located closer to industrial areas or with heavier vehicular traffic present higher levels of pollutants, and this translates into an increase in pediatric respiratory health problems in nearby areas.
- **Hypothesis D**: seasonal variations in pollutant levels correlate with peaks in the incidence of pediatric diseases, with certain pollutants showing a more prominent relationship in specific seasons of the year.

To evaluate the hypothesized link between air pollution and pediatric respiratory health, this study outlines a series of specific goals. These goals focus on analyzing relevant data sets and identifying potential associations:

- **Objective A**: analyze environmental pollution data from the Air Pollution Surveillance and Prediction Network (XVPCA).
- **Objective B**: analyze pediatric emergency room admission data provided by the Joan XXIII de Tarragona and Sant Joan de Reus hospitals from 2013 to 2023.
- **Objective C**: evaluate the associations and identify relevant patterns between environmental pollution and pediatric health.

Upon project completion, a comprehensive analysis will be conducted to evaluate the achievement of the research objectives and hypotheses. This analysis will involve a critical review of the data and the resulting findings. Key learnings will be extracted from the achieved results, highlighting any unexpected associations or discrepancies that may warrant further investigation. This process will not only confirm the validity of the initial hypotheses but also identify new avenues for future research in this field.

2 Theoretical framework

This chapter establishes the theoretical framework for investigating the relationship between air pollution and pediatric respiratory health in the "Camp de Tarragona". We will explore key areas: the characteristics of main air pollutants (Chapter 2.1), the Air Pollution Surveillance and Prediction Network (XVPCA) responsible for regional air quality monitoring (Chapter 2.2), relevant air quality legislation (Chapter 2.3), the established relationships between air pollutants and pediatric respiratory illnesses (Chapter 2.4), ethical considerations and approval by the Ethics Research Committee (Chapter 2.5), a comprehensive data management plan (Chapter 2.6), the concept of Big Data (Chapter 2.7), Python as the chosen programming language for data analysis (Chapter 2.8), and the benefits of using Google Colab, a cloud-based platform for running Python code (Chapter 2.9).

2.1 Main air pollutants

Industrial facilities and petrochemical processes are major contributors to air pollution, releasing a complex mixture of gaseous and particulate matter. Understanding the characteristics and origins of these pollutants is crucial for assessing their impact on human health, particularly in regions with high concentrations. This section will delve into the key air pollutants. We will explore their chemical composition, formation processes, and primary sources. This foundational knowledge will equip us to analyze potential health risks associated with these pollutants in the "Camp de Tarragona" and assess their impact on pediatric respiratory health later in Chapter 2.4.

Nitrogen oxides (NO_x) are a family of gaseous air pollutants formed by the reaction of nitrogen and oxygen during high-temperature combustion processes. These highly reactive gases are emitted primarily from vehicle exhaust, including automobiles, trucks, various non-road vehicles (construction equipment, boats…) and power plants, industrial boilers, cement kilns, and turbines [5]. It acts as a strong oxidizing agent, promoting reactions with volatile organic compounds (VOCs) that generate ozone particularly on hot summer days.

Among gaseous sulfur oxides (SO_x), sulfur dioxide (SO_2) is the pollutant of greatest concern due to its abundance. The primary source of SO_2 emissions is the combustion of fossil fuels in power plants and industrial facilities [6]. This highlights the significant role of human activities in contributing to atmospheric SO_2 levels. Additionally, vehicles and heavy equipment that burn fuels with high sulfur content can also contribute to SO_2 emissions.

Benzene (C_6H_6) is a volatile organic compound (VOC) readily released into the atmosphere. Traffic emissions are the primary source of outdoor benzene, with seasonal and meteorological variations impacting its concentration. Additionally, gas stations and various industries involved in coal, oil, natural gas, chemicals, and steel production contribute to ambient benzene levels [7].

Hydrogen sulfide (H_2S) is a colorless gas, often posing a hidden threat due to its odor being undetectable at low concentrations. The oil and natural gas industry is a major culprit, with emissions arising throughout the entire production chain, including extraction, processing, and even geothermal activity associated with these resources. Industrial activities also contribute significantly to H_2S emissions, with sources such as petrochemical plants, coke oven facilities, and kraft paper mills releasing this harmful gas [8].

Carbon monoxide (CO) is a colorless harmful gas at high concentrations. It is a byproduct of incomplete combustion, occurring whenever something is burned inefficiently. The primary outdoor sources of CO stem from vehicles and machinery powered by fossil fuels, such as cars, trucks, and other combustion engines [9].

Ozone (O_3) is a molecule formed by three oxygen atoms. High in the atmosphere, it forms the protective ozone layer. However, at ground level, ozone transforms into a harmful air pollutant due to reactions between precursor gases – nitrogen oxides (NO_x) emitted from burning fossil fuels and volatile organic compounds (VOCs) released from everyday products and industrial processes [10]. Sunlight fuels this atmospheric reaction, creating ozone smog that can be transported by wind, impacting areas far from its source.

Particulate matter (PM), a ubiquitous air pollutant, encompasses a mix of solid and liquid particles ranging from thick, visible specks to ultrafine particles invisible to the naked eye. Size is crucial, as our bodies can expel larger particles but fine and ultrafine ones infiltrate deep into the lungs and potentially the bloodstream. PM originates from both mechanical processes, like grinding down materials, and chemical processes, like fuel combustion that releases particles through gas reactions. Major sources include wildfires, wood burning, fossil fuel use in vehicles and power plants, and industrial activities [11].

These pollutants are particularly significant as they are the focus of data collected by the Air Pollution Surveillance and Prediction Network (XVPCA), explored in detail in the following chapter. Chapter 2.4 will then delve into the health issues associated with these pollutants.

2.2 Air Pollution Surveillance and Prediction Network (XVPCA)

The Air Pollution Surveillance and Prediction Network (XVPCA) is a critical component of environmental monitoring in Catalonia. Established in 1983, the XVPCA has evolved to encompass a comprehensive network of measurement points, analysis centers, and a central data repository. The network's core structure consists of strategically located measurement points where air samples are collected and analyzed to determine pollutant concentrations.

Data collected by the XVPCA follows a well-defined flow and validation process. Automated equipment at measurement points continuously monitors pollutant levels, generating data at regular intervals. Analysis centers retrieve this data from their assigned points through secure channels and perform rigorous validation checks before transmitting it to the central repository. The central data repository then receives, verifies, and archives the validated data, ensuring its integrity and accessibility for further analysis.

2.2. Air Pollution Surveillance and Prediction Network (XVPCA)

The XVPCA serves multiple crucial objectives. The primary goal is to monitor and evaluate air quality levels across Catalonia. This allows for the identification of areas with concerning pollution levels, tracking trends over time, and informing the development of targeted mitigation strategies. Public information is another vital objective. By providing citizens with accurate and up-to-date air quality data, the network empowers them to make informed decisions about their health and well-being. Additionally, air quality data is a valuable input for land-use planning, ensuring sustainable development practices and protecting vulnerable populations. The XVPCA also plays a vital role in regulatory compliance. Monitoring and reporting air quality data ensures adherence to environmental protection regulations and international standards.

The design and optimization of the XVPCA network are crucial for ensuring accurate and representative data collection. Several factors are considered when establishing measurement points, including the proximity to significant emission sources like industrial facilities and traffic corridors. Meteorological data is also collected to understand how weather patterns influence pollutant dispersion. The network considers geographical and topographical features to capture variations in air quality across different landscapes. Furthermore, measurement points are strategically placed in areas with high population density or sensitive receptors like schools and hospitals. The social demand for air quality monitoring and the potential impact on affected populations are also factored into the network's design. Lastly, special attention is given to ecologically or culturally significant areas that may be particularly vulnerable to air pollution.

By leveraging the Air Pollution Surveillance and Prediction Network (XVPCA) stations, we can obtain a comprehensive understanding of the pollutant types measured across the region. This data will be instrumental in investigating the coverage area for each hospital in relation to the specific pollutants of concern (Table 1). This will allow us to identify potential gaps and areas where pediatric populations might be disproportionately exposed to harmful air pollutants.

Station	Automatic	Manual	Hospital
Alcover (Mestral)	SO_2, NO_x, O_3, H_2S	-	Pius Hospital de Valls
Constantí (Gaudí)	C_6H_6, SO_2, NO_x, O_3, H_2S	C_6H_6, HCl, B(a)P, PM10, PM2.5, Metals	Hospital Universitari XXIII de Tarragona
El Morell (Deixalleria municipal)	-	C_6H_6	Hospital Universitari XXIII de Tarragona
La Canonja (deixalleria municipal)		C_6H_6, PM10	Hospital Universitari XXIII de Tarragona
Perafort (Puigdelfí)	C_6H_6, SO_2, NO_x, H_2S, PM10, PM2.5	C_6H_6	Hospital Universitari XXIII de Tarragona
Reus (el Tallapedra)	NO_x, O_3, CO, PM10, H_2S	C_6H_6	Hospital Universitari Sant Joan de Reus

Tarragona (Bonavista)	SO_2, NO_x, PM10, PM2.5, H_2S	-	Hospital Universitari XXIII de Tarragona
Tarragona (parc de la Ciutat)	SO_2, NO_x, O_3, CO, H_2S	C_6H_6	Hospital Universitari XXIII de Tarragona
Tarragona (Salut)	-	PM10, PM2.5, B(a)P	Hospital Universitari XXIII de Tarragona
Tarragona (Sant Salvador)	SO_2, NO_x, H_2S	C_6H_6	Hospital Universitari XXIII de Tarragona
Tarragona (Universitat Laboral)	SO_2, NO_x, H_2S	C_6H_6, PM10, PM2.5	Hospital Universitari XXIII de Tarragona
Vandellòs i l'Hospital de l'Infant (víver)	SO_2, NO_x	PM10	Hospital Universitari Sant Joan de Reus
Vila-seca (IES Vila-seca)	SO_2, NO_x, O_3, PM10, PM2.5, H_2S	C_6H_6, HCl, PM10, Metals	Hospital de Sant Pau i Santa Tecla
Vila-seca (la Pineda)	-	C_6H_6, PM10	Hospital de Sant Pau i Santa Tecla

Table 1. XVPCA Monitoring Stations: Pollutants Measured [12] **and Associated Hospitals.**

The XVPCA is a dynamic network that continuously adapts to evolving environmental conditions, technological advancements, and regulatory requirements. The network is regularly updated to incorporate new measurement techniques, expand its coverage, and address emerging pollutants. It also adheres to the latest European air quality standards and reporting protocols to ensure regulatory compliance. Data collected by the network is made available to the public through open data portals, promoting transparency and informed decision-making. Additionally, the XVPCA collaborates with research institutions to advance knowledge and develop innovative air quality monitoring solutions.

2.3 Legislation

Spain has a comprehensive legislative framework in place to regulate air pollutants and protect public health (Figure 3). The primary legislation governing air quality in Spain is Royal Decree 102/2011, of January 28, 2011, on improving air quality. This decree establishes emission limits for a variety of pollutants, as well as monitoring and reporting requirements for industrial facilities.

The decree has been amended several times since it was first enacted. The most recent amendments, which were made by Royal Decree 34/2023, of January 24, 2023, further tightened the emission limits for NO_2, PM2.5, and PM10.

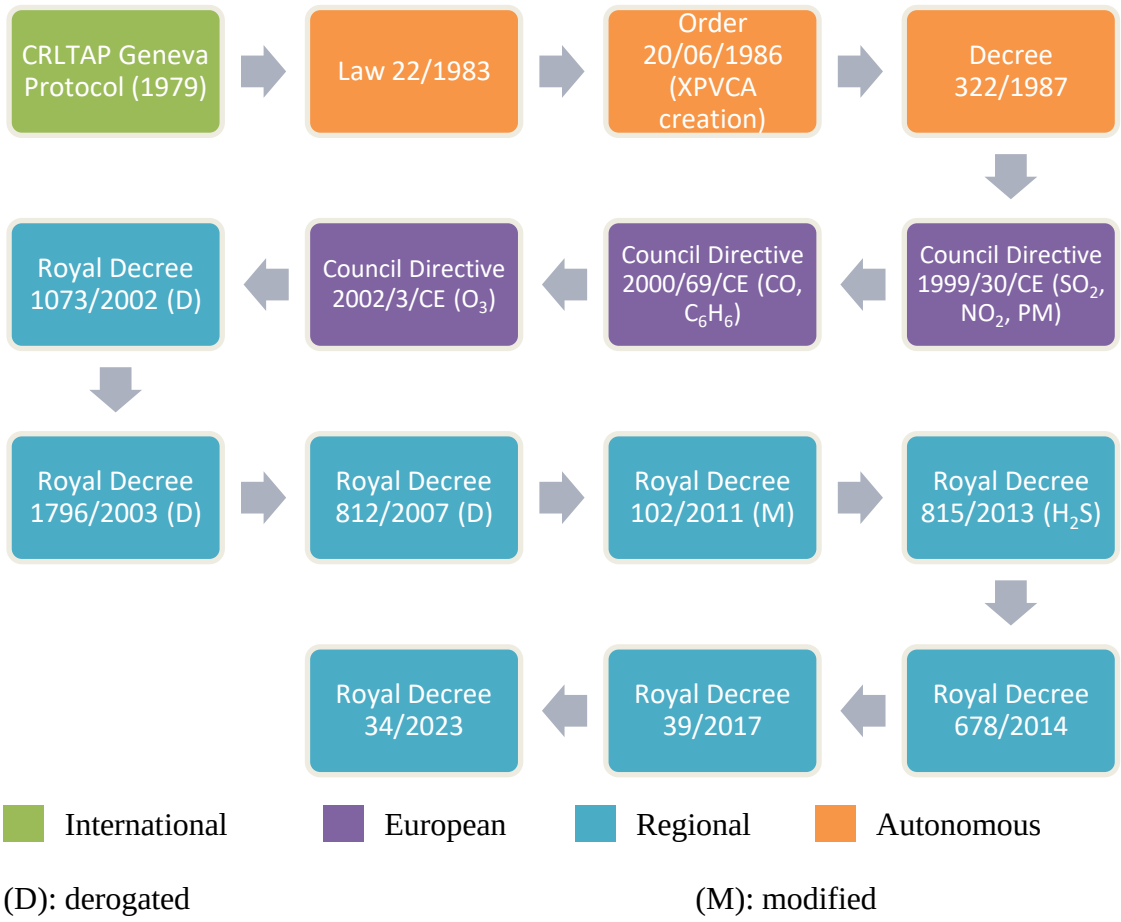


Figure 3. Air pollutants legislation in Spain.

Lately, the Spanish government approved the National Air Pollution Control Program (PNCCA) 2023-2030. The PNCCA focuses on reducing emissions of five key air pollutants: sulfur dioxide (SO_2), nitrogen oxides (NO_x), non-methane volatile organic compounds (NMVOCs), ammonia (NH_3), and fine particulate matter (PM2.5).

By reviewing all these regulations, we can identify the currently enforced limit values for the protection of human health (Table 2).

Pollutant	Temporary scope	Value
SO_2	1 year	20 µg/m³
NO_2	1 year	40 µg/m³
PM10	1 year	40 µg/m³
O_3	Maximum 8-hour average per day	120 µg/m³ (shall not exceed 25 times per year on average over a three-year period)
CO	Maximum 8-hour average per day	10 mg/m³
C_6H_6	1 year	10 mg/m³
PM2.5	1 year	25 µg/m³
H_2S	Daily average	40 µg/m³

Table 2. Air quality limits in Spain.

2.4 Pollution-related health diseases

Air pollution is a significant global health concern. Studies have consistently linked exposure to these pollutants with a variety of adverse health outcomes, particularly respiratory illnesses in children [13, 14]. Given the well-established link between air pollution and respiratory illness, we will specifically investigate the health consequences of air pollution on children's developing respiratory systems.

To ensure consistent and comparable data analysis, this research will use the International Classification of Diseases (ICD). Developed by the World Health Organization (WHO), the ICD provides a standardized system for coding diseases across healthcare settings globally [15]. This standardized coding allows researchers to accurately track and analyze pediatric respiratory illnesses across different populations and time periods. This facilitates robust comparisons and strengthens the generalizability of research findings.

To facilitate standardized analysis and result sharing, this study will leverage the pre-defined list of diseases provided by collaborating hospitals. These diseases can be readily mapped to the International Classification of Diseases (ICD-10) coding system (Table 3).

Most diseases provided by the hospitals are expected to have a single, corresponding ICD-10 code. However, some complex conditions, such as complicated pneumonia and respiratory superinfection, may involve multiple ICD-10 codes to capture the full clinical picture [16, 17].

ICD-10 Code	Disease
J00	Acute nasopharyngitis [common cold]
J12.0	Adenoviral pneumonia
J12.1	Respiratory syncytial virus pneumonia
J12.2	Parainfluenza virus pneumonia
J12.81	Pneumonia due to SARS-associated coronavirus
J12.89	Other viral pneumonia
J12.9	Viral pneumonia, unspecified
J13	Pneumonia due to Streptococcus pneumoniae
J15.0	Pneumonia due to Klebsiella pneumoniae
J15.20	Pneumonia due to staphylococcus, unspecified
J15.4	Pneumonia due to other streptococci
J15.5	Pneumonia due to Escherichia coli
J15.6	Pneumonia due to other Gram-negative bacteria
J15.7	Pneumonia due to Mycoplasma pneumoniae
J15.8	Pneumonia due to other specified bacteria
J15.9	Unspecified bacterial pneumonia
J16.8	Pneumonia due to other specified infectious organisms
J17	Pneumonia in diseases classified elsewhere
J18.0	Bronchopneumonia, unspecified organism
J18.2	Hypostatic pneumonia, unspecified organism
J18.9	Pneumonia, unspecified organism

J20.5	Acute bronchitis due to respiratory syncytial virus
J20.8	Acute bronchitis due to other specified organisms
J20.9	Acute bronchitis, unspecified
J21.0	Acute bronchiolitis due to respiratory syncytial virus
J21.8	Acute bronchiolitis due to other specified organisms
J21.9	Acute bronchiolitis, unspecified
J22	Unspecified acute lower respiratory infection
J30.1	Allergic rhinitis due to pollen
J30.5	Allergic rhinitis due to food
J30.81	Allergic rhinitis due to animal (cat) (dog) hair and dander
J30.89	Other allergic rhinitis
J30.9	Allergic rhinitis, unspecified
J31.0	Chronic rhinitis
J40	Bronchitis, not specified as acute or chronic
J41.8	Mixed simple and mucopurulent chronic bronchitis
J42	Unspecified chronic bronchitis
J45.20	Mild intermittent asthma, uncomplicated
J45.21	Mild intermittent asthma with (acute) exacerbation
J45.22	Mild intermittent asthma with status asthmaticus
J45.901	Unspecified asthma with (acute) exacerbation
J45.902	Unspecified asthma with status asthmaticus
J45.909	Unspecified asthma, uncomplicated
J45.990	Exercise induced bronchospasm
J45.991	Cough variant asthma
J91.8	Pleural effusion in other conditions classified elsewhere
J98.01	Acute bronchospasm

Table 3. Respiratory Diaseses and ICD-10 Codes for Analysis.

By leveraging the pre-defined diseases provided by collaborating hospitals and a comprehensive review of the scientific literature, we can establish robust associations between specific air pollutants emitted and the corresponding health effects on the developing respiratory systems of children (Table 4). This analysis will not only identify the pollutants but also pinpoint the targeted areas within the respiratory tract most impacted by each pollutant (Figure 4).

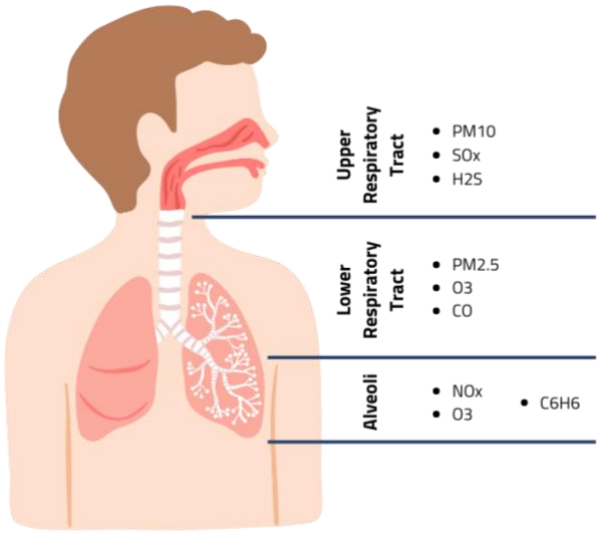


Figure 4. Air pollutants and the airway tract part they affect.

Air pollutant	Health effects
NO_x	Short-term exposure can worsen asthma symptoms, leading to coughing, wheezing, and difficulty breathing, potentially requiring emergency room visits or hospitalization. Chronic exposure to NO_2 may even contribute to the development of asthma and increase susceptibility to respiratory infections. Children, along with individuals with pre-existing respiratory conditions and the elderly, are especially vulnerable to the detrimental effects of NO_2. [18]
SO_x	Short-term spikes in SO_2 levels can trigger asthma attacks, causing wheezing, shortness of breath, and chest tightness (bronchoconstriction), especially during exercise when deeper, faster breaths carry SO_2 deeper into the lungs. Chronic exposure at high levels can also worsen respiratory symptoms and decrease lung function. These effects can lead to increased hospital admissions and emergency room visits, especially among vulnerable populations like children. [19]
C_6H_6	Inhaling high levels of benzene in the short term can cause a range of neurological effects like headaches, dizziness, and even unconsciousness. [20]
H_2S	Hydrogen sulfide exposure can trigger adverse health effects, particularly in children with asthma. Concentrations as low as 2 ppm for 30 minutes can irritate asthmatic airways. Generally, low levels (20-50 ppm) cause eye irritation, progressing to upper respiratory tract irritation at slightly higher concentrations. Prolonged exposure carries the risk of pulmonary edema (fluid buildup in the lungs). Since hydrogen sulfide is denser than air and settles at lower levels, children are more susceptible to inhaling higher concentrations due to their shorter stature. [21]
CO	Inhalation of carbon monoxide disrupts oxygen delivery throughout the body. High concentrations, particularly in enclosed spaces, can rapidly reduce the blood's ability to carry oxygen, leading to dizziness, confusion, unconsciousness, and even death. [22]

O_3	Ground-level ozone, a pollutant often elevated during hot, sunny days, poses a significant health risk. Individuals with asthma are especially vulnerable to the harmful effects of ozone exposure, which can exacerbate existing respiratory problems. [23]
PM2.5	Short-term exposure (up to 24 hours) to PM2.5 can be detrimental, leading to increased hospital admissions for heart or lung issues, asthma attacks, and respiratory symptoms. These effects are most pronounced in infants, children, and older adults with pre-existing respiratory or cardiovascular conditions. Long-term exposure (months to years) further amplifies the risks, potentially leading to premature death in individuals with chronic heart or lung diseases. [24]
PM10	Like PM2.5, short-term exposure to PM10 particles can exacerbate existing respiratory illnesses, particularly asthma and COPD. This can lead to a significant increase in hospitalizations and emergency department visits. While the long-term effects of PM10 require further investigation, some studies suggest a potential link to increased respiratory mortality. [24]

Table 4. Air pollutants and their health effects.

In conclusion, this subchapter has laid the groundwork for our research by establishing key relationships between air pollutants and the detrimental effects they have on children's developing respiratory systems (Figure 5).

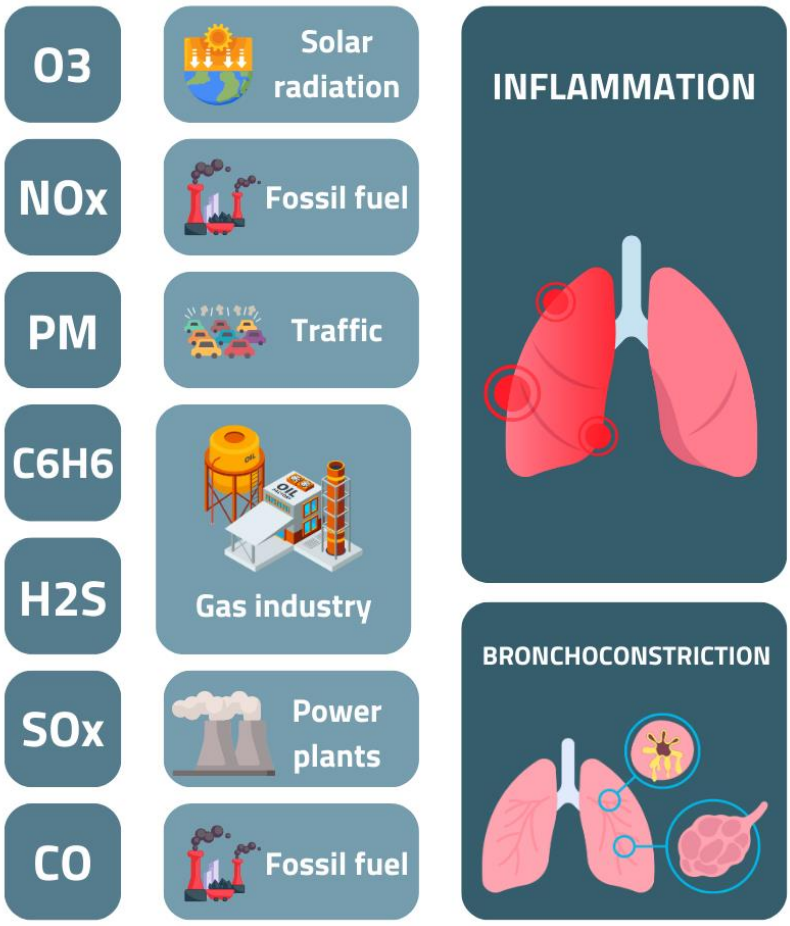


Figure 5. Air pollutants, their source and their main health effect on the respiratory system.

2.5 Ethics Research Committee (CEIm)

To ensure ethical conduct and participant protection, we obtained approval from the Ethics Research Committee (CEIm) at the "Institut d'Investigació Sanitària Pere Virgili" (IISPV). This independent committee, comprised of healthcare professionals and laypeople, reviews research projects to safeguard the rights, safety, and well-being of research participants. Their focus aligns with Good Clinical Practice guidelines, ensuring adherence to ethical, methodological, and legal standards.

Our application to the CEIm included a detailed proposal outlining the study. This proposal included key documents such as the study evaluation request form, data protection questionnaire, clinical research protocol, commitment letters from researchers and collaborators, and conformity statements from hospital directors and the IISPV. Additionally, a declaration of activities with minors (for each center), economic memory (for each hospital), and essay synopsis (only HUSJR) were submitted. All documents addressed the research question, data requirements, data handling procedures, background information, study objectives, hypotheses, methodology, statistical methods, ethical considerations, and data confidentiality measures. These documents were signed by the principal investigator for each participating center and the study collaborators.

The data request encompassed electronic health records of patients aged 0-14 years who visited the emergency department of the collaborating hospitals between January 1st, 2013, and December 31st, 2023. The specific diseases of interest included: acute bronchitis, bronchiolitis, bronchospasm, pneumonia (both acute and complicated), respiratory superinfection, asthma, acute rhinitis, acute rhinopharyngitis, pleural effusion, and lung abscess. In addition to diseases, the data request included patients' birthdates, genders, places of residence, and visit dates.

Committee meetings are held on the last Thursday of every month, with proposals due by the 10th. Our project received approval from the CEIm on March 21st. This approval signified the ethical acceptability of our research and allowed us to proceed with data collection from participating hospitals.

2.6 Data management plan

A data management plan (DMP) is an essential component of the research process, ensuring the organization, preservation, and accessibility of research data throughout the lifecycle of a project. The DMP for this research project adheres to the guidelines and recommendations set forth by the Digital Curation Centre (DCC). By outlining procedures for data collection, storage, analysis, sharing, and long-term preservation, this DMP aims to promote responsible data stewardship and maximize the value of your research data for future scientific inquiry.

Plan Overview

Title: CASE_TGN: Study of the relationship between air pollution and pediatric health in the Camp de Tarragona

Creator: Dídac Roda Pitarg

Principal Investigator: Noelia Ramírez González

Data Manager: Dídac Roda Pitarg

Affiliation: Other

Template: DCC Template

Project abstract:

This study aims to investigate the relationship between air quality and pediatric emergency department visits for respiratory problems in the Camp de Tarragona region. With a time frame of 2013 to 2023, the study will address this crucial issue to understand the implications of air pollution on children's health.

To achieve this objective, data on pediatric emergency department admissions for respiratory illnesses at the Sant Joan de Reus Hospital will be analyzed.

The research will be based on two main data sources: air quality data collected from the Air Quality Surveillance and Prediction Network and clinical data from pediatric patients who have been admitted to the emergency department with respiratory problems, with the collaboration of the hospital's pediatric unit.

ID: 151628

Start date: 25-03-2024

End date: 30-06-2024

Last modified: 20-05-2024

CASE_TGN: STUDY OF THE RELATIONSHIP BETWEEN AIR POLLUTION AND PEDIATRIC HEALTH IN THE CAMP DE TARRAGONA

DATA COLLECTION

What data will you collect or create?

The primary data source will be pseudoanonymized electronic health records (EHRs) obtained from participating hospitals. This data will encompass demographics (age, gender, postal code), diseases of respiratory illnesses using ICD-10 codes (e.g., asthma, pneumonia), and emergency department visit dates between January 1st, 2013, and December 31st, 2023. The anticipated data volume will be estimated based on the retrieved children's health records. All data will be converted into a standardized format, such as comma-separated values (.csv), to ensure compatibility and ease of analysis. This format prioritizes long-term usability, sharing, and archiving capabilities. The plan will be reviewed to explore the potential for incorporating existing relevant datasets from other studies, maximizing data utilization and comprehensiveness.

The collected data will be securely stored on the "Universitat Rovira i Virgili" OneDrive server with restricted access granted only to authorized personnel involved in the research project. A regular backup plan will be implemented to ensure data security.

Following completion of the research project, a suitable data repository that adheres to the FAIR principles (Findable, Accessible, Interoperable, Reusable) will be identified for data archiving by each collaborating hospital. The data will be pseudoanonymized according to relevant data protection regulations (GDPR). Comprehensive data documentation will be prepared, clearly outlining the data content, format, variables, and any coding schemes used. A data sharing agreement will be established, specifying access terms and any potential restrictions on data access.

Throughout the research process, all data will be pseudoanonymized to safeguard participant privacy. The project will comply with all relevant ethical regulations and data protection laws.

A version control system (Git) will be implemented to track any modifications made to the data throughout the research project. This ensures data integrity and allows for reverting to previous versions if necessary.

This data management plan will be a living document and will be updated as the research progresses or if any significant changes occur in data collection, storage, or sharing procedures.

How will the data be collected or created?

Electronic health records (EHRs) will be the primary source of data for this project. We will collaborate with participating hospitals to obtain pseudoanonymized EHR data for children aged 0-14 who visited their emergency departments between January 1st, 2013, and December 31st, 2023. The data request will comply with relevant data protection regulations and encompass demographics (age, gender, postal code), diseases using ICD-10 codes (focusing on respiratory illnesses), and emergency department visit dates.

We will also explore incorporating additional data sources. This includes air quality data from local monitoring stations.

A standardized folder structure with descriptive names (e.g., EHRs_2013, AirQualityData_2020) will be used to organize the collected data. Subfolders may be created for further categorization within each data type. To ensure data integrity and allow for reverting to previous versions, if necessary, a version control system (Git) will be implemented to track any modifications made to the data throughout the research process. Each data version will be clearly labeled with a version number and modification details.

Several measures will be taken to guarantee data quality. The obtained EHR data will be reviewed for missing values, inconsistencies, and outliers, with cleaning procedures documented for transparency. Standardized data capture procedures will be defined and used whenever possible during data collection from various sources to minimize errors during integration. Data entry will be validated through double-checking and potentially using range checks to identify implausible values. Finally, a data dictionary will be created to document data variables, definitions, coding schemes, and any data transformations applied.

DOCUMENTATION AND METADATA

What documentation and metadata will accompany the data?

To ensure the long-term understandability and reusability of the data for future researchers, this data management plan outlines the creation of comprehensive documentation and metadata to accompany the dataset.

A central data dictionary will be established, defining each data variable. This will include descriptions, coding schemes used (ICD-10 codes), and any data transformations applied. A detailed methodology report will be created, outlining the research methodology, data collection procedures, inclusion/exclusion criteria for participants, and any data quality control measures implemented.

The metadata will adhere to community standards to promote discoverability and interoperability of the data. The Dublin Core Standard will be used to capture basic information about the data, including title, creators, subject, description, and dates. Data-specific metadata will also be included, encompassing variable names and descriptions (corresponding to the data dictionary), data types, units of measurement, file formats, and software used for data collection and processing.

The data dictionary and methodology report will be created as separate electronic documents. Metadata will be embedded within the data files themselves using appropriate file format functionalities. A "readme" text file will be included with the data package, summarizing key information about the data, documentation, and metadata.

ETHICS AND LEGAL COMPLIANCE

How will you manage any ethical issues?

This data management plan prioritizes ethical considerations and ensures participant privacy throughout the research process.

In this project, the exemption of informed consent from the patients is requested to the Etics Research Committee (CEIm) because it is a retrospective study, the data of which are obtained from pre-existing records and no new interventions are performed.

All data obtained from EHRs will be pseudoanonymized by the collaborating hospitals before analysis and storage. This includes removing direct identifiers such as names and addresses. Postal codes will not be anonymized to maintain geographic relevance. The pseudoanonymized data will be stored on the “Universitat Rovira i Virgili” OneDrive server with restricted access granted only to authorized personnel involved in the research project. Secure data transfer protocols will be implemented whenever data needs to be transferred between locations.

Following anonymization, the data may be deposited in a data repository that adheres to FAIR principles and relevant data protection regulations (GDPR). Access to the data will be controlled through a data sharing agreement outlining terms of access and potential restrictions.

How will you manage copyright and Intellectual Property Rights (IPR) issues?

The ownership of the pseudoanonymized data collected from EHRs will be determined by agreements with participating hospitals. In the absence of such agreements, the data copyright will reside with each participating hospital. Any third-party data sources, such as air quality data, will be incorporated following adherence to the data provider's terms and conditions. Restrictions on reuse specified by the data providers will be considered when establishing the data sharing agreement for the overall research dataset.

The data may be made available for reuse through a data repository adhering to the FAIR principles. A data sharing agreement will be established, outlining a specific license for data access and reuse. This license will likely be a Creative Commons license that permits non-commercial use, attribution, and potentially share-alike requirements. Restrictions on data access may be implemented for a limited embargo period to allow for initial publication of research findings.

STORAGE AND BACKUP

How will the data be stored and backed up during the research?

To ensure robust security and reliable data access, the anonymized data will be stored on a OneDrive server provided by the "Universitat Rovira i Virgili". This managed storage solution adheres to institutional data security policies and industry best practices. Storing data on personal laptops, computer hard drives, or external storage devices will be avoided to minimize risks of data loss.

Automatic backup procedures will be implemented to create regular backups of the data. These backups will be stored on a separate secure server within the university network. This redundancy ensures data recovery in case of a primary storage failure.

How will you manage access and security?

Access to the pseudoanonymized data will be restricted to authorized personnel directly involved in the research project. Each authorized user will be required to have unique login credentials for accessing the data storage server. This ensures accountability and prevents unauthorized access. The data storage server adheres to institutional data security policies and industry best practices. This ensures the implementation of appropriate security measures to safeguard data confidentiality, integrity, and availability. Additionally, all data transfers will be encrypted to further protect sensitive information during transmission.

For collaborators requiring access to the data for research purposes, secure data sharing procedures will be established. This may involve depositing the data in a trusted data repository that enforces access controls. Alternatively, collaborators may be granted temporary access to a secure university server environment. Regardless of the method, data access will be subject to data sharing agreements outlining permitted uses and potential restrictions.

This data management plan discourages the use of personal email accounts or non-university approved cloud storage services (OneDrive) for data transfers due to potential security risks.

SELECTION AND PRESERVATION

Which data are of long-term value and should be retained, shared, and/or preserved?

This data management plan identifies the following data for long-term retention, sharing, and preservation:

- Anonymized data extracted from electronic health records (EHRs) relevant to the research question. This includes demographic information (age, gender, zip code), diseases (coded using ICD-10), and air quality exposure data (linked by station code and time period).
- A data dictionary detailing variable definitions and coding schemes used.
- A methodology report outlining the research design, data collection procedures, and data quality control measures.

These datasets hold potential value for future research on the long-term health effects of air pollution on children. Researchers may use the data to validate our findings, explore new research questions, or conduct comparative analyses.

The pseudoanonymized data and associated documentation will be retained within the research group for the time set by current legislation. Beyond the minimum retention period, data preservation may be pursued through deposit in a trusted data repository that adheres to the FAIR principles (Findable, Accessible, Interoperable, Reusable). The chosen repository will provide long-term data storage, access control mechanisms, and data preservation practices to ensure data integrity over time.

Prior to data deposit, the data will be prepared for sharing according to the data repository's guidelines. This may involve formatting adjustments, and creation of detailed metadata to facilitate data discovery and reuse. Data access will be controlled through data sharing agreements outlining terms of use and potential restrictions.

What is the long-term preservation plan for the dataset?

Beyond the minimum retention period mandated by institutional or legal requirements, this data management plan prioritizes the long-term preservation of the anonymized dataset and associated documentation.

The data may be deposited in a trusted data repository that adheres to the FAIR principles (Findable, Accessible, Interoperable, Reusable). When selecting a repository, we will consider factors such as domain expertise to ensure discoverability by relevant researchers, data preservation practices to guarantee long-term integrity and accessibility, and access controls to manage data sharing according to agreements. Repository fees and any required data preparation efforts will also be factored into the selection process. We

will consult with the chosen repository to ensure alignment with their data deposit procedures, metadata standards, and any associated costs.

Prior to data deposit, the pseudoanonymized dataset will be prepared according to the repository's guidelines. This may involve converting data to a well-documented, non-proprietary format suitable for long-term storage, and creating comprehensive metadata using a widely accepted standard to facilitate data discovery and understanding by future researchers. The data dictionary and methodology report will be converted into machine-readable formats for archival purposes.

This data management plan acknowledges potential costs associated with long-term data preservation, such as repository data storage fees and staff time dedicated to data preparation for archiving. These costs will be factored into project budgets and funding requests where applicable.

DATA SHARING

How will you share the data?

The pseudoanonymized data and associated documentation may be deposited in a trusted data repository that adheres to the FAIR principles (Findable, Accessible, Interoperable, Reusable). This approach ensures discoverability through the repository's search functions and interoperability with other research data due to adherence to data standards. Data access will be controlled through data sharing agreements established with the repository. These agreements may outline conditions for data access, such as requiring researchers to register for an account and accept terms of use. Restrictions on data access may be implemented for a limited embargo period to allow for initial publication of research findings.

Following data deposit, a persistent identifier (DOI) will be obtained for the dataset. This unique identifier will facilitate ongoing data discovery and citation within the research community, enabling researchers to properly credit the source of the data when reused in future studies.

The anonymized format and detailed metadata documentation will minimize restrictions on data reuse, allowing researchers to explore new research questions or conduct secondary analyses.

Are any restrictions on data sharing required?

Data users will be required to acknowledge the data source and adhere to the terms of the data sharing agreement established with the data repository. These agreements typically outline conditions for data access and reuse, helping to ensure responsible data stewardship practices.

RESPONSIBILITIES AND RESOURCES

Who will be responsible for data management?

The Principal Investigator (PI) bears overall responsibility for implementing and adhering to this data management plan. This includes ensuring data collection adheres to ethical research protocols and participant consent procedures. The PI will also appoint a data management team member responsible for day-to-day data management activities.

A designated team member will be responsible for the day-to-day data management activities. These activities include data quality control, data storage and security procedures, and metadata creation. This team member will also coordinate with the chosen data repository to ensure successful data deposit and archiving.

This data management plan acknowledges that the hospitals "Joan XXIII de Tarragona" and "Sant Joan de Reus" are responsible for the data as Data Controllers, according to relevant data protection regulations (GDPR). However, the research team will be responsible for data management activities throughout the research lifecycle, as described above.

What resources will you require to deliver your plan?

Personnel with dedicated time and effort will be crucial. The Principal Investigator (PI) will oversee data management plan implementation, while a designated staff member within the research team will handle day-to-day activities. These activities include data quality control, data storage and security procedures, metadata creation, and communication with the chosen data repository.

Technology and infrastructure needs include secure data storage on the university server throughout the project duration. Depending on the data and chosen repository, data formatting, and conversion into a repository-compatible format may be necessary.

The budget will factor in potential data repository fees associated with data deposit and long-term archiving. Training for research staff on data management best practices, including data security procedures and data preparation for archiving, will also be provided.

Justification for these expenses lies in the importance of responsible data management practices. Effective data management ensures data quality, security, and long-term preservation, ultimately contributing to the credibility and reproducibility of research findings. By dedicating these resources, we aim to maximize the value of the research data for the scientific community.

Finally, if the chosen data repository imposes user fees for data access or download, this will be clearly communicated to potential data users. Researchers can then factor these costs into their own research budgets when considering data reuse.

2.7 Big Data

In recent years, the ability to collect and analyze vast amounts of data, often referred to as "big data", has revolutionized numerous fields (Figure 6). This includes the healthcare sector, where large, complex datasets – structured and unstructured – hold immense potential for uncovering hidden patterns.

Figure 6. Data generated in only 60 seconds in different sectors (adopted from [25]**).**

While the concept of storing and analyzing large datasets has existed for decades, the term "big data" exploded in the early 2000s. This shift reflected the emergence of data that challenged traditional processing methods due to its sheer volume, rapid growth, and inherent complexity. Industry analyst Doug Laney famously captured this phenomenon with the "three V's" of big data:

- **Volume**. The amount of data organizations collect has skyrocketed. Transactions, data from smart devices and industrial equipment, videos, images, and social media posts all contribute. Traditionally, storing such vast quantities would have been very expensive. However, advancements in data lakes, Hadoop, and cloud storage have made managing this data much more manageable.
- **Velocity**. The rise of the Internet of Things (IoT) has lead to an era of unprecedented data speed. Information streams into businesses at an ever-increasing pace, demanding real-time or near-real-time processing. Technologies like RFID tags, sensors, and smart meters generate massive amounts of data that require immediate attention.
- **Variety**. Data comes in an assorted array of formats, no longer confined to the structured, numeric data found in traditional databases. Today, we deal with unstructured text documents, emails, videos, audio files, and financial transactions – all valuable sources of information.

While the original definition of big data focused on the three V's, the field has evolved to encompass additional characteristics. In recent years, two more V's have emerged as crucial aspects of big data:

- **Variability**. Beyond the pure speed and diversity of data, big data also presents a challenge in its unpredictable nature. Data streams can fluctuate significantly, with trends emerging and fading rapidly. Businesses must be adaptable, able to detect real-time social media shifts and manage surges in data volume triggered by daily, seasonal, or event-based fluctuations.
- **Veracity**. The volume of data, often originating from various sources, introduces another hurdle: data quality, or veracity. Effectively linking, matching, cleaning, and transforming data across various systems becomes a complex task. Businesses need robust strategies to establish connections, hierarchies, and ensure accurate data linkages to avoid data integrity issues and maintain control.

The true value of big data lies not just in its vastness, but in the transformative insights it unlocks through analysis. By harnessing data from diverse sources, organizations can achieve significant advancements across various aspects. Data-driven insights can optimize resource allocation and utilization, leading to enhanced resource management. Identifying and eliminating inefficiencies in processes through big data analysis translates to improved operational efficiency. Additionally, leveraging data to understand customer needs and preferences fuels product development innovation. Unveiling hidden market trends and customer behavior patterns through big data empowers businesses to discover new revenue streams and growth opportunities. Ultimately, big data empowers data-driven decision making, allowing organizations to make informed and strategic choices.

These aren't just theoretical benefits. Big data analysis is making a significant impact across various industries. Predictive maintenance, for example, analyzes data patterns to identify potential equipment failures before they occur, preventing costly downtime. Fraud detection systems leverage big data to analyze data patterns and detect fraudulent activities in real-time. In the healthcare sector, big data analysis contributes to precision medicine by enabling the analysis of medical data to personalize treatment plans and improve patient outcomes. Dynamic risk management utilizes big data to quickly recalculate risk profiles based on real-time market changes, and big data also plays a role in enhancing AI models. By training AI models with vast amounts of data, big data improves their accuracy and adaptability.

Effectively harnessing big data requires careful consideration of its complex flow across diverse locations, sources, systems, and stakeholders. This "big data fabric" encompasses not only traditional structured data, but also unstructured and semi-structured information. To navigate this landscape, businesses can leverage a five-step approach (Figure 7).

Figure 7. Big data workflow.

2.8 Python

Since its debut in 1991, Python has established itself as a leading interpreted programming language, alongside Perl and Ruby. Its popularity went through the roof with the emergence of web development frameworks like Django. But Python's capabilities extend far beyond scripting. Due to historical and cultural factors, Python fomented a big scientific computing and data analysis community. Over the past decade,

it has transitioned from a regular scientific language to a cornerstone for data science, machine learning, and general software development in both academia and industry.

When it comes to data analysis and interactive visualization, Python naturally invites comparisons with other open-source and commercial tools like R or Matlab. However, Python's recent advancements in library support have been instrumental in its dominance in data analysis tasks. Libraries like pandas and pyspark provide powerful functionalities specifically designed for data manipulation and machine learning. When combined with Python's overall strength in general-purpose software engineering, it becomes a compelling choice as the primary language for building data-driven applications. Leveraging its strengths in data analysis and machine learning, Python has been chosen as the primary programming language for this research work.

2.9 Google Colaboratory

The rise of cloud computing has revolutionized the field of deep learning. There are some platforms that offer a compelling solution for researchers by providing readily available hardware resources. Leading cloud providers like Amazon and Google Cloud offer access to powerful GPUs and pre-configured deep learning environments, often on a pay-per-use basis. This flexibility and scalability are particularly attractive for research projects that require significant computational power without the upfront costs of dedicated hardware.

Building upon this trend, Google introduced Colaboratory (Colab), a cloud service specifically designed to facilitate machine learning education and research [26]. Colab provides a fully configured runtime environment equipped with leading AI libraries and access to powerful GPUs. Moreover, it seamlessly integrates with Google Drive accounts and provides free access to these resources. This makes Colab a particularly attractive option for researchers with limited budgets, offering a free and accessible platform to conduct deep learning research as in this research work.

3 Data Analysis

This chapter explores the analysis of the two key datasets: air pollution and clinical admissions data. We will begin with a descriptive analysis of each dataset, examining trends and patterns within air pollutants (Chapter 3.1) and health outcomes (Chapter 3.2). Following this, Chapter 3.3 will analyze both datasets together to investigate potential correlations between air pollution and hospital admissions.

Considering the large size of our datasets, we will leverage Apache Spark (PySpark) for data manipulation and analysis. PySpark is a popular distributed processing framework specifically designed for handling Big Data efficiently [27]. Additionally, we will use Pandas and Matplotlib for effective data visualization.

To ensure transparency and reproducibility, preprocessed datasets, annotated Python notebooks documenting the data cleaning and analysis procedures, and the final cleaned datasets will be archived in a private GitHub repository. This approach facilitates version control, allowing for the tracking of changes made to the data throughout the research process.

3.1 Air pollution analysis

This air pollution dataset is openly available through the Catalonia government's "Dades Obertes" portal, as discussed in Chapter 2.2. The XVPCA differentiates between pollutants measured automatically and those requiring manual intervention. This distinction is reflected in the portal, where the data is separated into automatic and manual datasets. To obtain a comprehensive picture of air pollution, we will acquire both datasets and merge them following necessary transformations.

While the Catalonia government's "Dades Obertes" portal allows direct data viewing, we will leverage its Application Programming Interface (API) for efficient data acquisition. This approach enables us to filter the data based on our specific needs. Through the portal, we will select the desired monitoring stations, years, and pollutants. Subsequently, we will utilize the retrieved API endpoint to download the filtered data directly in CSV format, facilitating further analysis.

We will use Python to interact with the API endpoints provided by the "Dades Obertes" portal. This approach allows us to retrieve both the automatic and manual air pollution datasets efficiently. Once retrieved, we will convert the data from its original format (CSV) into PySpark dataframes. Following data acquisition, we will perform necessary preprocessing steps to prepare the automatic and manual datasets for subsequent merging.

A key distinction between the automatic and manual datasets lies in their sampling frequency. Automatic monitoring stations collect data every hour, resulting in a dataframe with one column for each hour of the day. Conversely, manual stations only provide one measurement per day, reflected in the dataframe's structure with columns representing each day of the month.

To address the difference in sampling frequency between the automatic and manual datasets, we will employ a data transformation strategy aligned with our initial analysis goals. Since we begin with a yearly descriptive analysis (Figure 8), we will calculate the average concentration of each pollutant for each year and monitoring station. This process creates a consistent time resolution across both datasets, enabling a meaningful merge. Following the merge, we can then proceed with data visualization, such as plotting the average pollutant levels per year for each monitoring station.

Figure 8. Air pollution dataset analysis process.

While a yearly analysis provides a high-level overview, it may conceal seasonal variations in air pollution. To address this limitation, we will delve deeper with a monthly analysis. This approach allows us to identify potential trends and fluctuations in pollutant concentrations across different seasons. To effectively visualize these spatial and temporal patterns, we will leverage the Plotly library.

To prepare the data for monthly analysis, we will perform aggregations on the PySpark dataframes. This will involve grouping the data by pollutant, monitoring station, year, and month. Subsequently, we will calculate the mean of the hourly measurements for the automatic stations and the daily measurements for the manual stations. This process ensures consistent time resolution across the merged dataset. As our analysis involves spatial visualization, we will retain the latitude and longitude information already provided by the XVPCA for each data point. This will allow us to effectively map pollution levels across the monitoring stations.

Following the data aggregation within PySpark, we will transition the DataFrame to pandas for further analysis and plotting. Once converted, we will address missing values that might arise due to days without measurements in the automatic dataset or months lacking data in the manual dataset. Finally, we will sort the resulting pandas dataframe by year and month. This chronological organization ensures that the data on the map is displayed in a temporally meaningful sequence for viewers.

Prior to data visualization, we will integrate the established Catalan Air Quality Index (ICQA) into our analysis (Figure 9). The ICQA provides a standardized classification system for pollutant concentrations, categorizing them into levels ranging from “Good” to “Extremely Unfavourable”. This classification scheme allows us to translate raw pollutant concentrations into easily interpretable air quality categories. Subsequently, we can leverage these classifications to create informative map-based visualizations for each pollutant. By employing animation with time as the frame and ICQA levels represented by color, we can effectively communicate spatial and temporal variations in air quality across the monitoring stations.

Contaminants (base temporal de càlcul)		Nivell de ICQA (concentració en µg/m³, excepte pel CO que són en mg/m³)					
		Bona	Raonablement bona	Regular	Desfavorable	Molt desfavorable	Extremadament desfavorable
Diòxid de nitrogen (NO_2)	1-hora	0-40	41-90	91-120	121-230	231-340	>340
Partícules en suspensió PM10	24-hores	0-20	21-40	41-50	51-100	101-150	>150
Partícules en suspensió PM2,5	24-hores	0-10	11-20	21-25	26-50	51-75	>75
Ozó troposfèric (O_3)	8-horari	0-50	51-100	101-130	131-240	241-380	>380
Diòxid de sofre (SO_2)	1-hora	0-100	101-200	201-350	351-500	501-750	>750
Monòxid de carboni (CO)	1-hora	0-2	3-5	6-10	11-20	21-50	>50
Benzè (C_6H_6)	1-hora	0-5	6-10	11-20	21-50	51-100	>100
Sulfur d'hidrogen (H_2S)	1-hora	0-25	26-50	51-100	101-200	201-500	>500

Figure 9. ICQA levels per contaminant [28].

While spatial visualization offers valuable insights into geographic patterns, our analysis extends further to explore the temporal distribution of pollutant levels throughout the year. We will also compute the mean concentration, which serves as a central tendency measure, and the standard deviation, which quantifies the variability in pollutant levels across the year. This comprehensive approach allows us to not only identify spatial variations in air quality but also understand the temporal fluctuations and inherent variability experienced at each monitoring station.

To facilitate the analysis in Chapter 3.3, which explores the correlation between air pollution and clinical data, we will export a subset of the air pollution dataset. This subset will include daily pollutant concentrations for each pollutant and city. By converting this data into a comma-separated values (CSV) file, we ensure compatibility with the tools used in the subsequent chapter.

3.2 Clinical analysis

While our initial request to the Ethics Research Commitee (CEIm) on March 10th, 2024, included data from both Joan XXIII Hospital (HUJ23) in Tarragona and Sant Joan de Reus Hospital (HUSJR), we ultimately received data only for HUSJR as of May 31st, 2024. To ensure timely completion of the project, we decided to proceed with the available HUSJR data. This decision acknowledges the reduced dataset size for the correlation analysis and model training compared to the originally planned scope.

The clinical admissions data encompasses a ten-year period, which presented a challenge due to a software change within the hospital system. This software transition resulted in separate datasets with varying schemas. To address this complexity, we received the data in a single Excel file containing four distinct sheets, each with a different data structure.

To identify and handle missing location data, we first employed data visualization techniques. By plotting the distribution of missing values across each dataset, we were able to pinpoint columns with significant information gaps, particularly the city of residence field. Following this initial assessment, we opted for a listwise deletion approach. This strategy removes entire rows containing missing city residence data. While this approach can lead to data loss, it ensures consistency and facilitates further analysis focusing on patients with complete location information.

An additional challenge arose due to the first dataset using ICD-9 diagnostic codes, while the subsequent datasets employed ICD-10. To ensure consistent coding across all datasets and facilitate meaningful analysis, we undertook the manual mapping of ICD-9 diseases to their corresponding ICD-10 equivalents. This process leveraged the online resource icd10data.com, which offers a conversion tool to streamline the task. While manual data manipulation can be time-consuming, it was necessary to establish a standardized coding system for diseases across the entire dataset.

Three out of the four datasets directly provide the city name. However, the first dataset lacks this field and instead includes postal codes. To ensure consistent location data across all datasets, we leveraged an open-source dataset mapping Spanish postal codes to corresponding city names.

Following the aforementioned data standardization steps, we conducted additional data cleaning and transformation tasks to prepare the four dataframes for merging. This process involved addressing remaining inconsistencies, formatting issues, or duplicates.

The data cleaning and merging processes resulted in a comprehensive clinical admissions dataset. This dataset comprises 37,668 rows, with each row representing an individual patient record. The key data points captured for each record include date of admission, patient gender, age, city of residence, and diagnostic code. This rich dataset provides a solid foundation for further analysis to explore potential correlations between air pollution and hospital admissions in Chapter 3.3.

We will now refine the analysis scope to focus on a specific geographic area of interest, the “Camp de Tarragona” region. This region encompasses six sub-regions: Alt Camp, Baix Camp, Baix Penedès, Conca de Barberà, Priorat, and Tarragonès. To achieve this targeted selection, we will leverage the Application Programming Interface (API) provided by the Catalonia Statistics Institute (IDESCAT). This approach allows us to efficiently filter the data based on precise geographical boundaries. The resulting dataset will comprise approximately 34,817 patient records, representing a reduction of nearly 8% compared to the initial dataset.

Following the selection of patients residing within the “Camp de Tarragona” area, we will start with a descriptive analysis to gain foundational insights into the characteristics of the study population. This analysis will encompass two key aspects: demographics and diseases. Firstly, we will examine the distribution of patients by gender and age. This will provide a clear picture of the patient population's composition.

Secondly, we will identify the three most prevalent diseases within the dataset. Understanding these common diseases will establish a context for subsequent analysis exploring potential relationships between air pollution and specific health outcomes.

3.3 Combined analysis

Building upon the data cleaning and preparation steps outlined in the preceding subchapters, we can now start the main analytical phase of this research. The initial stage involves uploading the two cleaned CSV files containing the air pollution and clinical admissions data.

The XVPCA air quality monitoring network not encompass all city locations within the "Camp de Tarragona" area. To address this potential limitation and ensure a more comprehensive analysis, we will employ the Geopy Python package to identify the nearest monitoring station for each city in the dataset. We will establish an arbitrary maximum allowable distance threshold of 10 kilometers between a city and its designated monitoring station (Code 1). Any data points exceeding this distance threshold will be excluded from the analysis. This approach acknowledges the limitations of the XVPCA network while attempting to maximize data inclusion based on spatial proximity. For instance, the Priorat region lacks a monitoring station entirely. Consequently, the majority of data points from this region are likely to be excluded due to the distance criterion.

```
all_cities =
daily_clean_data.select("municipio_nombre").distinct().collect()

assigned_stations = {}

all_stations =
daily_pollution.select("nom_estacio").distinct().collect()

for city in all_cities:
  location_city2 = geolocator.geocode(city)

  closest_distance = float('inf')
  closest_station = None

  lat_long_city2 = (location_city2.latitude, location_city2.longitude)

  for estacio in all_stations:
    location_city1 = geolocator.geocode(estacio)

    lat_long_city1 = (location_city1.latitude,
location_city1.longitude)
    distance = GD(lat_long_city1, lat_long_city2).km

    if distance <= 10 and distance < closest_distance:
        closest_distance = distance
        closest_station = estacio
```

```
        assigned_stations[city] = estacio
```

Code 1. Function to find nearest station to a city.

Following the identification of the nearest monitoring station for each city, we will proceed with patient assignment. This will involve a join operation on the two dataframes, linking each patient record to its corresponding closest air quality monitoring station based on their home city location. Subsequently, we can merge the resulting dataframe with the air quality dataset. The merge criteria will be based on both date and monitoring station identifier. It is anticipated that a small number of patient records may not be assigned a monitoring station due to exceeding the distance threshold or residing in areas without coverage. These data points, identifiable by a null value in the corresponding station field, will be excluded through listwise deletion to ensure data integrity for subsequent analyses, giving a final dataset containing 62,852 rows.

To study the relationship between the number of pediatric admissions due to respiratory exacerbations and air pollutant concentrations, we will focus on four pollutants (NO_x, SO_2, PM10, and PM2.5) as these are consistently measured by the majority of XVPCA stations. Spearman's rank correlation and Pearson's correlation coefficient will be employed to assess the strength and direction of these relationships. Additionally, we will compare the correlations obtained before and after log-transformation of the pollutant concentration data.

To comprehensively evaluate the relationship between air pollutant concentrations and hospital admissions, we will analyze four distinct scenarios:

1. All admissions: this analysis will encompass admissions for all diseases.
2. J00 admissions (acute nasopharyngitis): we will specifically focus on admissions with the ICD-10 code J00, representing acute nasopharyngitis (commonly known as the common cold).
3. Pre-lockdown admissions (all diagnoses): this scenario will limit the analysis to admissions prior to the COVID-19 Spain lockdown in March 2020.
4. Pre-lockdown J00 admissions: this analysis will combine the focus on J00 admissions with the pre-lockdown timeframe.

To conclude the analysis, we will develop a data model for prediction employing machine learning techniques. Specifically, we will utilize Support Vector Regression (SVR) and Partial Least Squares Regression (PLSR) models. Both models will be trained on 80% of the data, with the remaining 20% reserved for testing and evaluating their predictive capabilities.

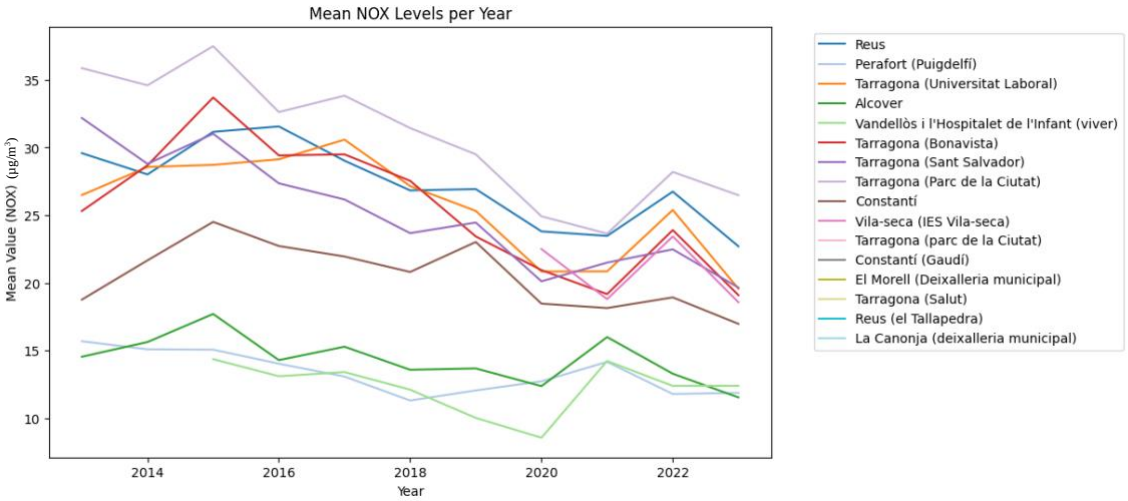
4 Results

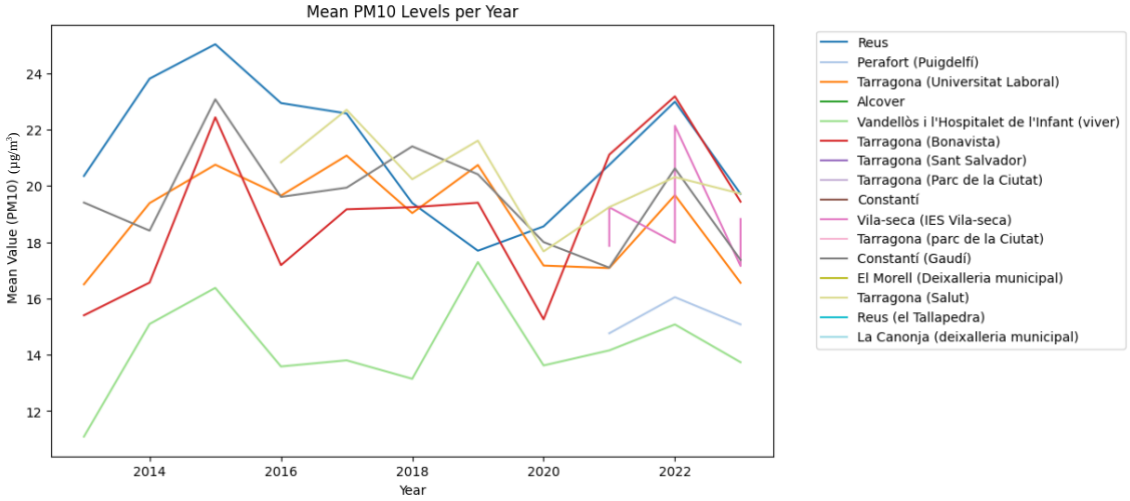
Building upon the methodological framework outlined in Chapter 3, this chapter presents the key findings derived from the data analysis. We will systematically explore the results, interpreting their significance in the context of air quality in the "Camp de Tarragona" area.

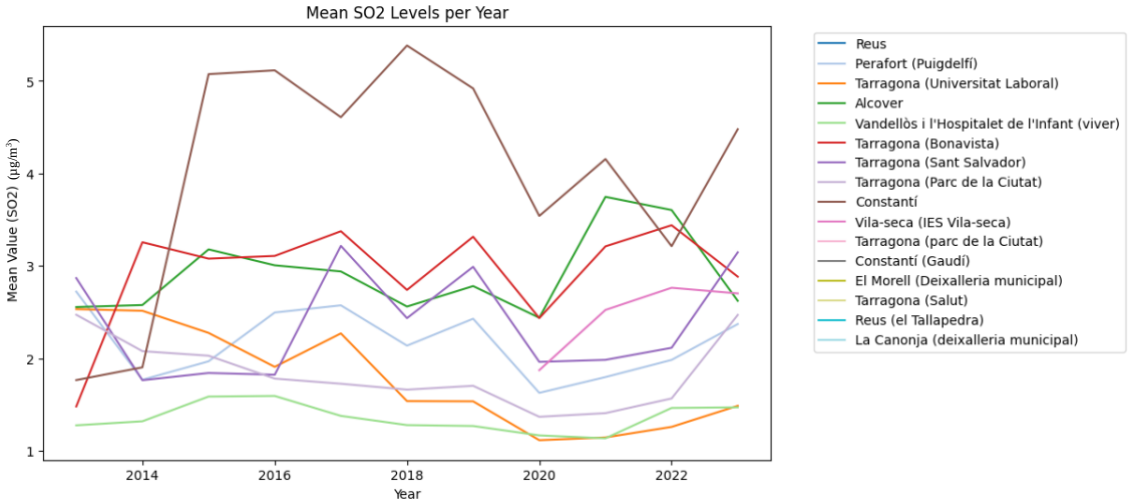
4.1 Air pollution

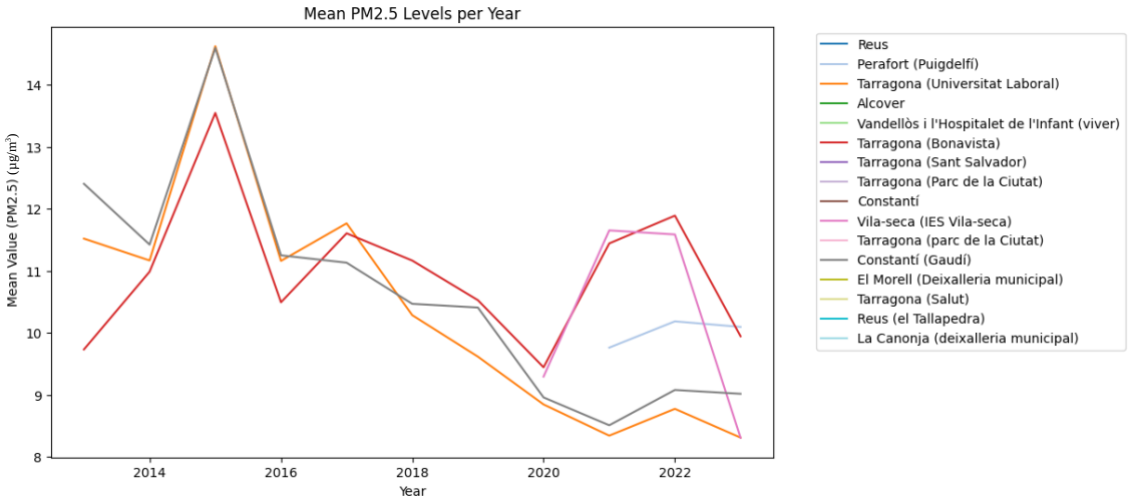
The initial descriptive analysis of the air pollution dataset (Appendix B) reveals spatial variations in pollutant concentrations. The table shows mean values and standard deviations for each pollutant at each monitoring station across the study period (2013-2023). This data suggests potentially higher pollution levels in urban areas compared to less populated locations. For instance, stations in Tarragona, Reus, and Vila-seca exhibit higher values compared to stations in Alcover or Vandellòs. Notably, La Canonja station appears to have the highest benzene concentrations. This finding could be linked to its proximity to the major industrial zone, warranting further investigation. It's important to note that measurement units vary across pollutants, with µg/m^3 used for all except CO, which is measured in mg/m^3.

To complement Appendix B, we will use graphical representations to visualize trends in pollutant concentrations across the study period (Figure 10). These visualizations will depict the mean values of pollutants for each monitoring station over time (2013-2023).

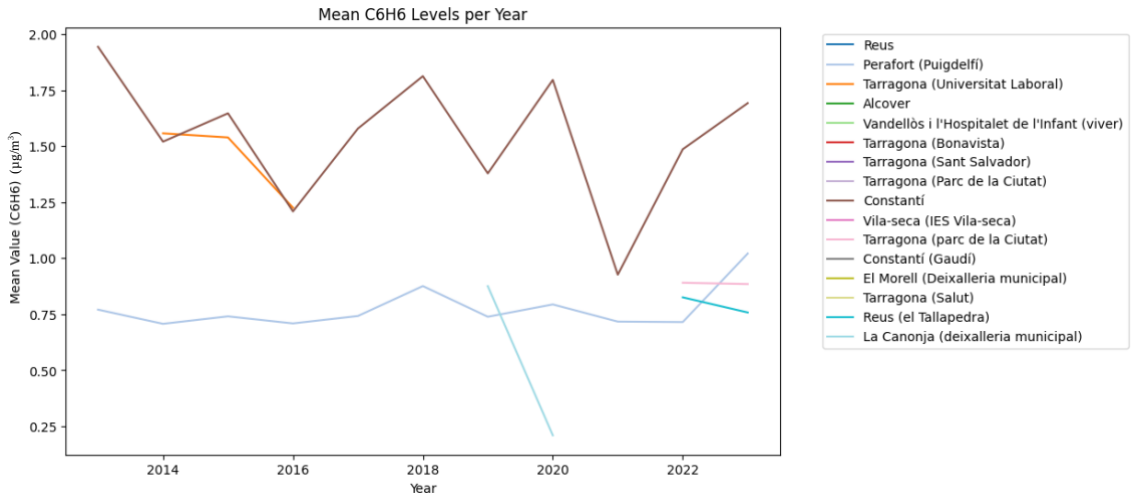


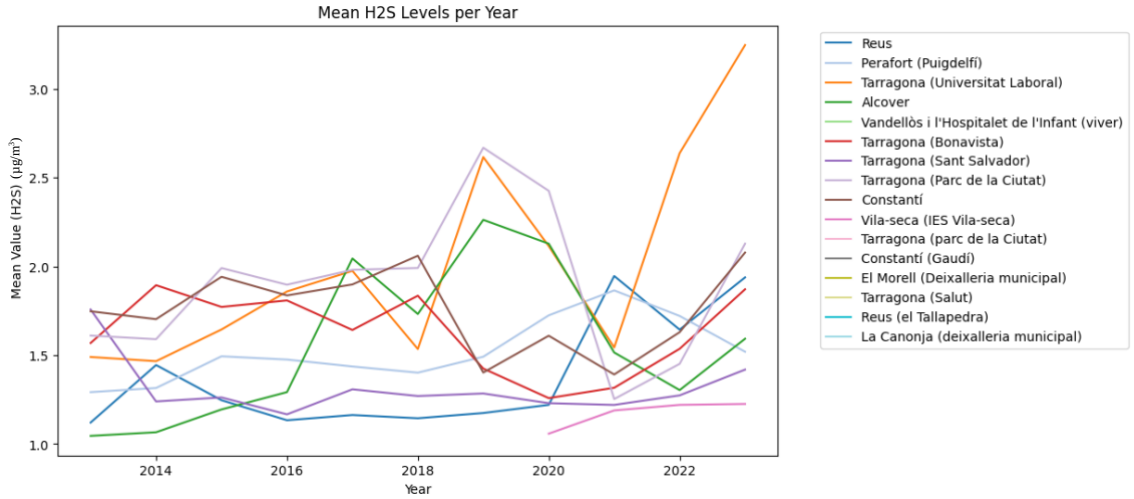

Mean PM10 Levels per Year
Mean Value (PM10) (µg/m³)
Year
Reus
Perafort (Puigdelfí)
Tarragona (Universitat Laboral)
Alcover
Vandellòs i l'Hospitalet de l'Infant (viver)
Tarragona (Bonavista)
Tarragona (Sant Salvador)
Tarragona (Parc de la Ciutat)
Constantí
Vila-seca (IES Vila-seca)
Tarragona (parc de la Ciutat)
Constantí (Gaudí)
El Morell (Deixalleria municipal)
Tarragona (Salut)
Reus (el Tallapedra)
La Canonja (deixalleria municipal)

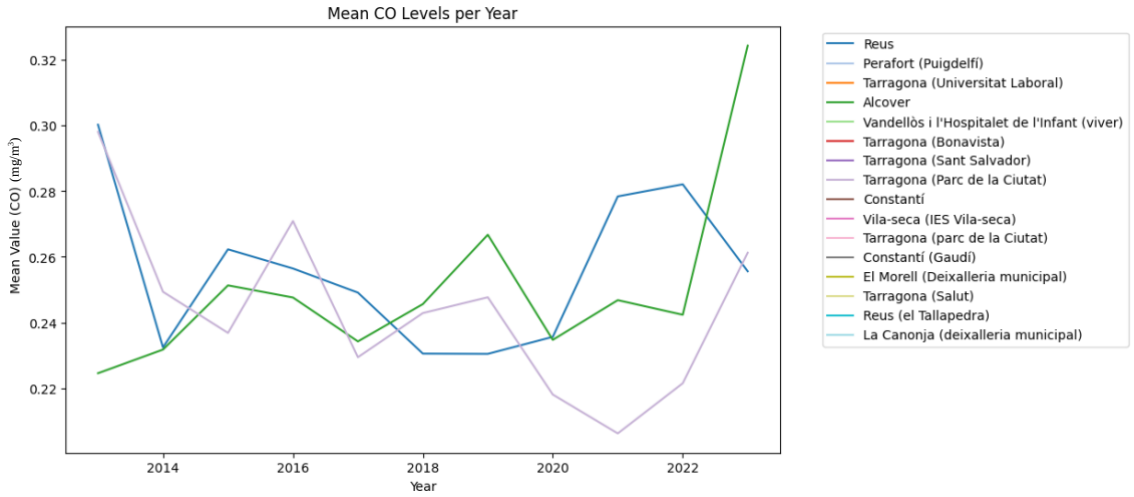

Mean SO2 Levels per Year
Mean Value (SO2) (µg/m³)
Year
Reus
Perafort (Puigdelfí)
Tarragona (Universitat Laboral)
Alcover
Vandellòs i l'Hospitalet de l'Infant (viver)
Tarragona (Bonavista)
Tarragona (Sant Salvador)
Tarragona (Parc de la Ciutat)
Constantí
Vila-seca (IES Vila-seca)
Tarragona (parc de la Ciutat)
Constantí (Gaudí)
El Morell (Deixalleria municipal)
Tarragona (Salut)
Reus (el Tallapedra)
La Canonja (deixalleria municipal)


Mean PM2.5 Levels per Year
Mean Value (PM2.5) (µg/m³)
Year
Reus
Perafort (Puigdelfí)
Tarragona (Universitat Laboral)
Alcover
Vandellòs i l'Hospitalet de l'Infant (viver)
Tarragona (Bonavista)
Tarragona (Sant Salvador)
Tarragona (Parc de la Ciutat)
Constantí
Vila-seca (IES Vila-seca)
Tarragona (parc de la Ciutat)
Constantí (Gaudí)
El Morell (Deixalleria municipal)
Tarragona (Salut)
Reus (el Tallapedra)
La Canonja (deixalleria municipal)

4.1. Air pollution







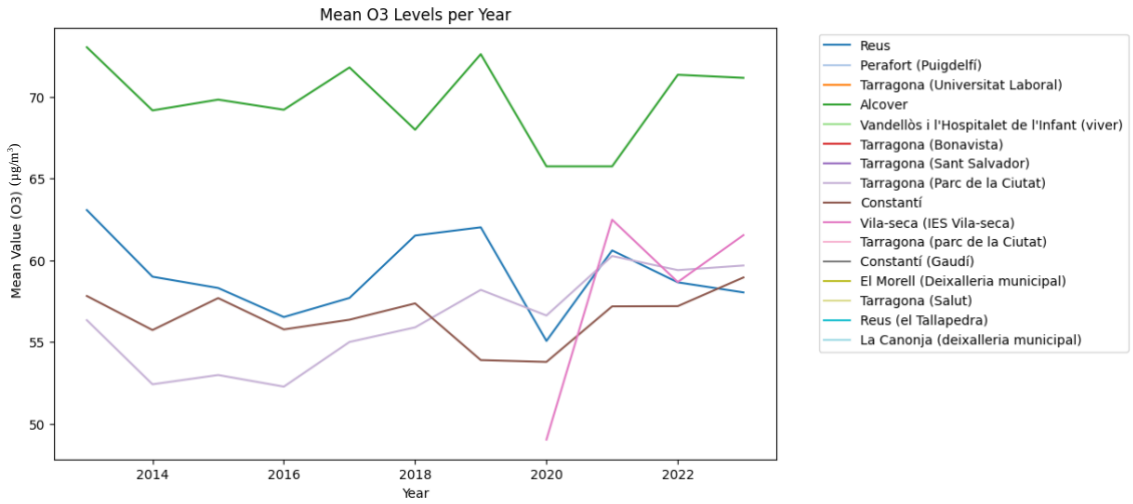


Figure 10. Plots of the mean concentration of pollutants per year and monitoring station.

The data suggests spatial variations in pollutant concentrations across monitoring stations. Notably, Vandellòs exhibits consistently low levels for all measured pollutants. Conversely, Alcover shows generally low values except for hydrogen sulfide, carbon monoxide, and ozone, where it has the highest concentrations. This observation warrants further investigation into potential local sources of these specific pollutants in Alcover.

Furthermore, a noteworthy decrease in pollutant concentrations is observed in 2020. This decline could be associated with the COVID-19 lockdown in Spain, which led to reduced activity in the petrochemical complex and road transport.

Following the descriptive analysis, we will examine the distribution of air quality classifications based on the Catalan Air Quality Index (ICQA) for each monitoring station, year, and pollutant (Appendix C). Stations such as Constantí, Reus, Tarragona (Bonavista), Tarragona (Parc de la Ciutat), Tarragona (Salut), and Tarragona (Universitat Laboral) are exhibit a higher frequency of unfavorable classifications based on the preliminary observations of pollutant concentrations.

To conclude the descriptive analysis of the air pollution dataset, we can explore monthly variations in pollutant concentrations across monitoring stations (animated plots, download the file and open it in a browser). This analysis involve map visualizations that depict the distribution of pollutant concentrations for each month and station over the study period.

4.2 Clinical data

Having explored the air quality data through descriptive analysis, we now shift our focus to the pediatric hospital admission data for respiratory exacerbations at Sant Joan de Reus hospital.

Following the data cleaning procedures outlined in Chapter 3.2, the final dataset for pediatric admissions due to respiratory exacerbations at Sant Joan de Reus hospital

comprises n = 34,817 records. This sample exhibits a relatively balanced distribution between male and female patients (Figure 11).

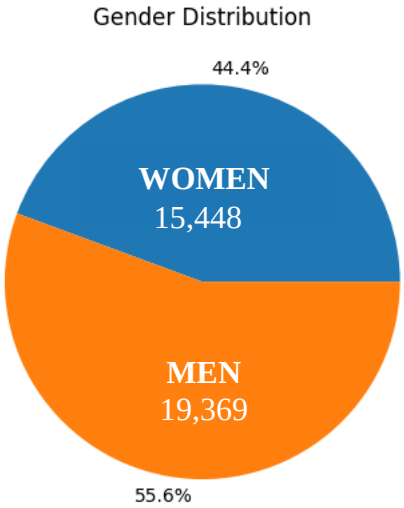


Figure 11. Gender distribution in the clinical dataset.

The analysis of pediatric admissions for respiratory exacerbations reveals a clear age distribution. The highest frequency of admissions occurred in children under one year of age, followed by a gradual decrease in admissions throughout childhood up to the age of 16 (Figure 12).

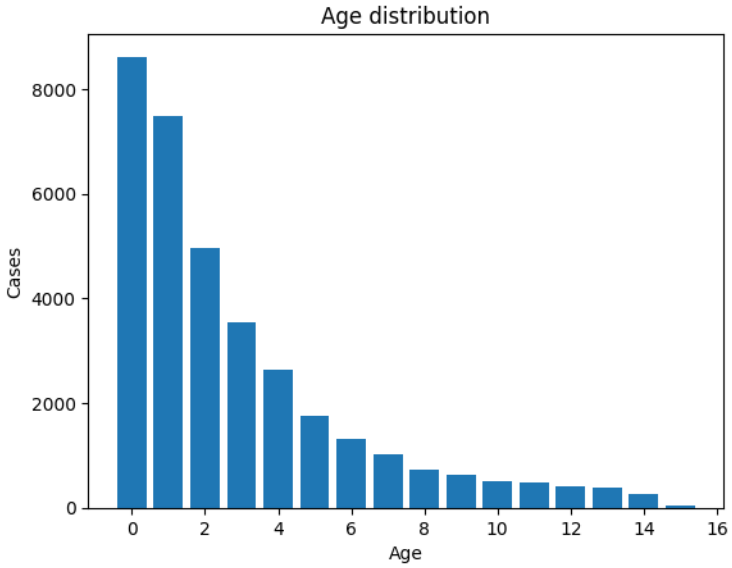


Figure 12. Age distribution in the clinical dataset.

Given the presence of nearly fifty different diagnoses within the pediatric admission dataset, we will focus on the three most frequently occurring illnesses. Analysis revealed that acute nasopharyngitis emerged as the most prevalent disease, followed by bronchitis and bronchiolitis (Figure 13).

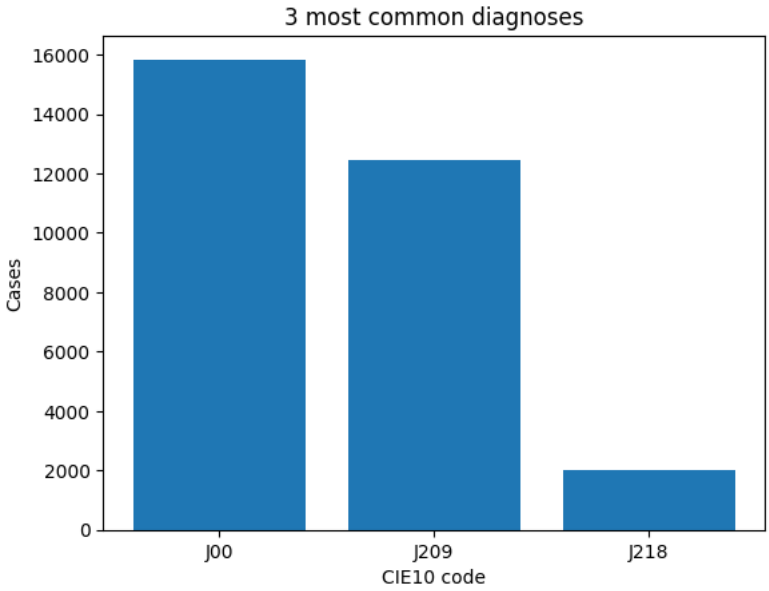
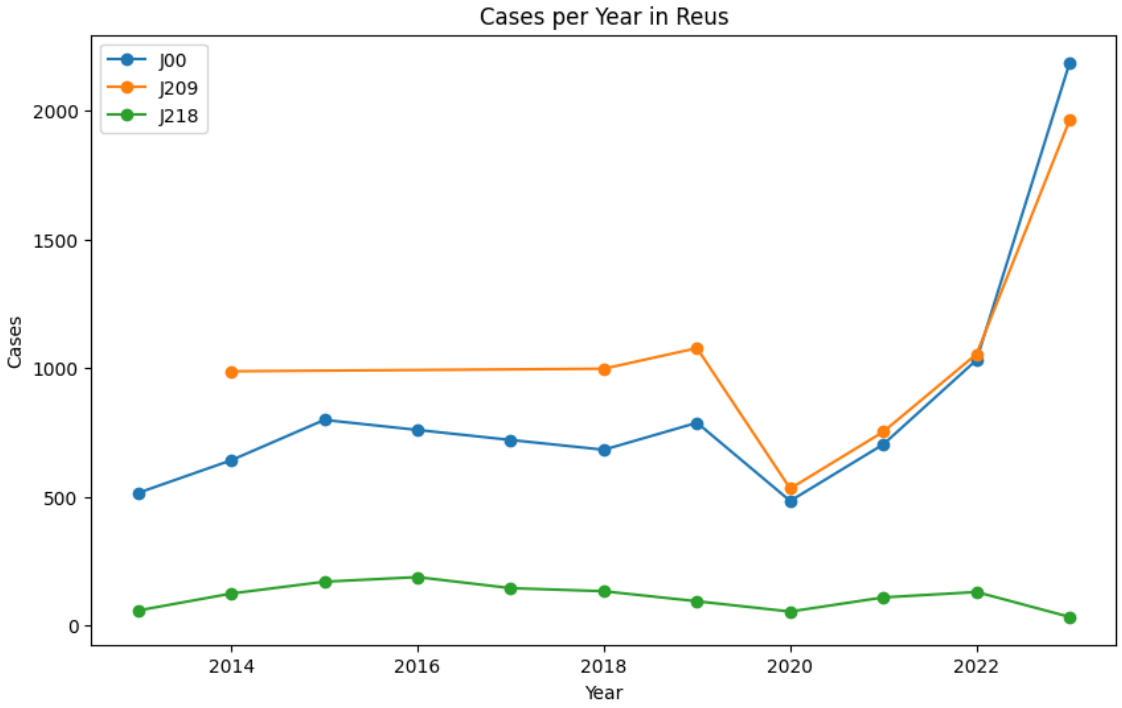


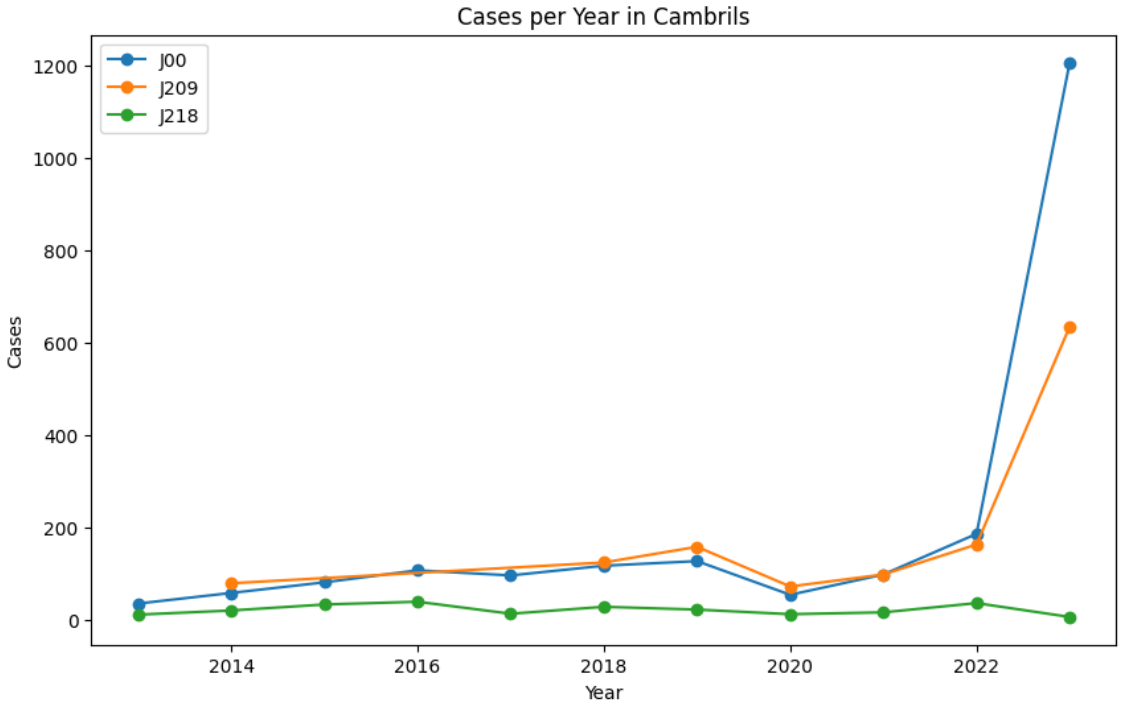
Figure 13. Three most frequent diseases in the clinical dataset.

To conclude the descriptive analysis of the clinical dataset, we will explore the geographic distribution of hospital admissions across patients' home cities. This analysis will focus on admissions for the three most prevalent respiratory illnesses identified earlier (acute nasopharyngitis, bronchitis, and bronchiolitis) (Figure 14).

The time series plot reveals a noteworthy decrease in admissions for all three prevalent diseases during the year 2020. This decline potentially coincides with the implementation of mandatory face mask use due to the COVID-19 pandemic. Public health measures, such as mask-wearing, are known to reduce transmission of respiratory viruses. Further investigation is needed to determine the extent to which the pandemic and associated control measures influenced these observed trends.

Conversely, the data shows an increase in admissions during 2023. While the reasons for this rise are unclear at this stage, further investigation is warranted to explore potential contributing factors.


Cases per Year in Reus
J00
J209
J218
2000
1500
1000
500
0
Cases
2014
2016
2018
2020
2022
Year


Cases per Year in Cambrils
J00
J209
J218
1200
1000
800
600
400
200
0
Cases
2014
2016
2018
2020
2022
Year

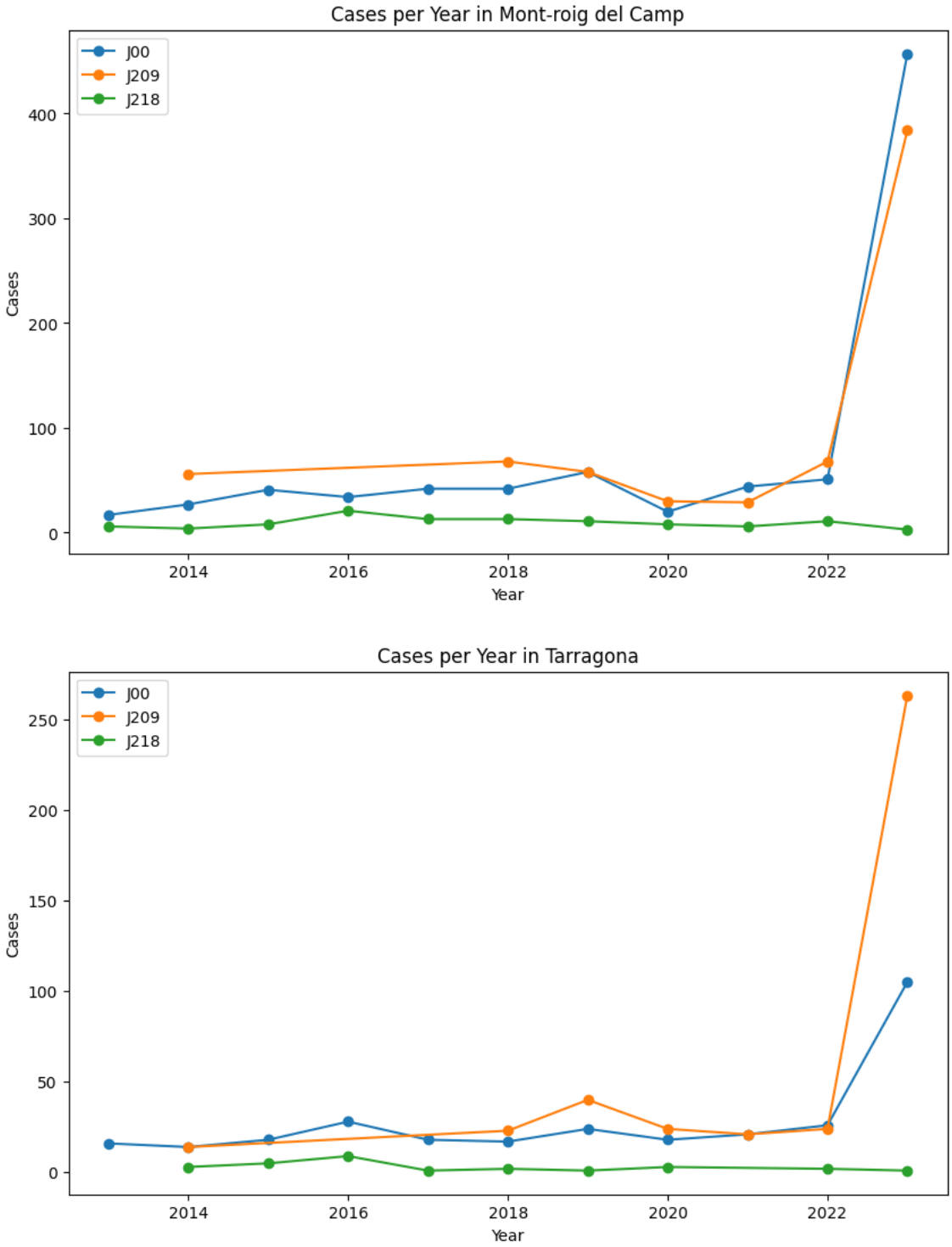


Figure 14. Number of cases per disease in different cities of the Camp de Tarragona area.

4.3 Correlation and Machine Learning

Following the descriptive analysis of both air quality and pediatric admission data, we employed Spearman's rank correlation coefficient to assess the potential association between air pollutant concentrations and hospital admissions for respiratory exacerbations. While the initial results using daily data did not reveal strong correlations, we opted to explore alternative approaches to refine the analysis. This included examining correlations based on weekly averaged pollutant concentrations and comparing the results obtained with Spearman's rank to those derived from Pearson's correlation coefficient (Appendix D). These additional analyses aimed to identify a more robust approach for investigating potential relationships between air quality and respiratory health outcomes.

The Spearman's rank correlation coefficients revealed different degrees of association between air pollutant concentrations and hospital admissions for respiratory exacerbations. Notably, SO_2 exhibited no significant correlations across any analysis scenarios. PM2.5 displayed a weak positive correlation with total admissions when using daily data (both pre- and post-log transformation). PM10 also showed a weak positive correlation with total admissions for daily time slots pre-lockdown, but only after log transformation.

Notably, NO_x emerged as the pollutant with the strongest correlations. However, the strength of this association depended on the analysis approach. A weak positive correlation was observed between daily NO_x levels (pre-log transformation) and total admissions, as well as admissions for J00 (acute nasopharyngitis) pre-lockdown (post-log transformation). Notably, the most significant correlation coefficient (Spearman's rank = 0.5) was found between weekly NOx concentrations (pre-lockdown) and weekly admissions for all diseases after log transformation (Figure 15).

These findings suggest that the relationship between air quality and respiratory admissions might be complex and influenced by factors such as time aggregation (daily vs. weekly) and specific disease categories. Additionally, the comparison between Spearman's rank and Pearson's correlation coefficients indicates that Spearman's rank may be more suitable for analyzing potentially non-normally distributed data.

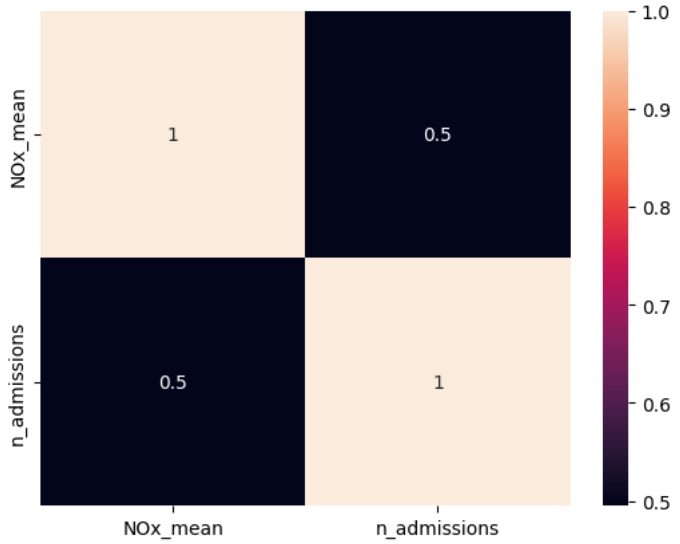


Figure 15. Spearman's correlation between NO_x concentration and clinical admissions before the COVID-19 lockdown.

Having explored potential associations between air quality and pediatric admissions through correlation analysis, we proceeded to use machine learning algorithms. Our goal was to develop a predictive model capable of estimating the number of hospital admissions for respiratory exacerbations based on air pollutant concentrations. The model considered the mean concentration of the pollutant to which patients were likely exposed, determined by the closest monitoring station to their home address. This approach aimed to move beyond basic correlations and explore the potential for machine learning to predict future admissions based on air quality data.

Following the application of machine learning algorithms, we evaluated the performance of two models: Support Vector Regression (SVR) and Partial Least Squares (PLS) regression. The data was divided into an 80% training set and a 20% testing set to assess the models' generalization. The SVR model achieved an accuracy of 0.18, while the PLS model achieved an accuracy of 0.25.

It's important to note that these accuracy values indicate limited predictive power in the current models. While an accuracy of 0.25 suggests the PLS model might predict the number of admissions for one week within a month on NO_x exposure, further refinement is necessary for robust prediction (Figure 16).

We further investigated the potential for improved model performance by exploring various configurations. This included trying with daily data, incorporating other pollutants as input features, and testing different weighting schemes for training and testing datasets. However, these attempts resulted in models with lower accuracy compared to the initial approach using weekly NO_x exposure data.

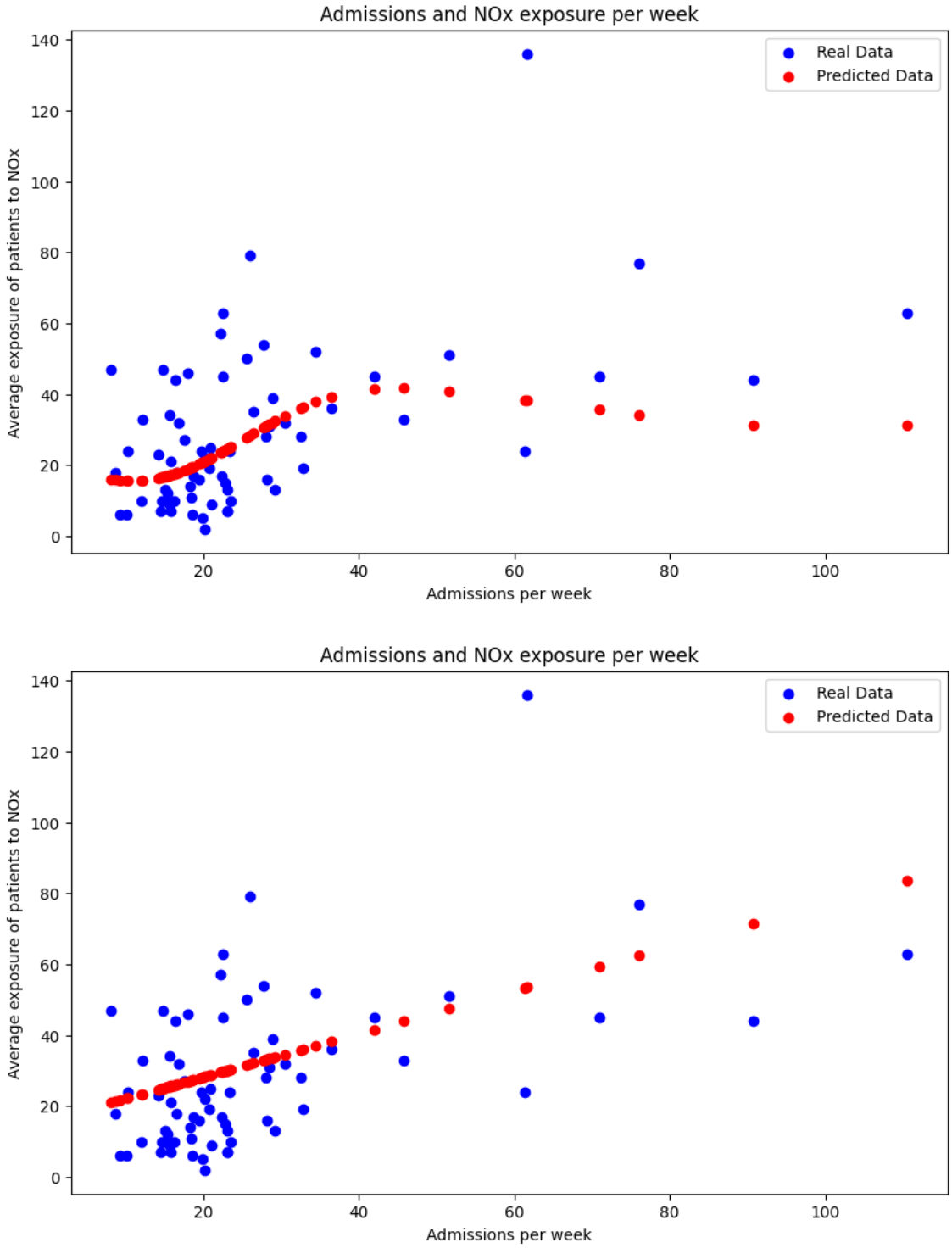


Figure 16. Number of weekly admissions vs. predicted number of weekly admissions.

5 Conclusions

This study investigated the potential link between air pollution exposure and pediatric respiratory health in the industrial region of Camp de Tarragona, Spain. By analyzing air quality data from the Air Pollution Surveillance and Prediction Network (XVPCA) alongside pediatric admission data for respiratory illnesses from Sant Joan de Reus Hospital, the research aimed to shed light on potential associations and their impact on children's health.

Descriptive analysis revealed spatial variations in air pollutant concentrations across monitoring stations. Higher levels were observed in urban areas compared to rural locations. A notable decrease in pollutant concentrations was observed in 2020, likely coinciding with the COVID-19 lockdown restrictions.

Analysis of pediatric admissions data identified acute nasopharyngitis, bronchitis, and bronchiolitis as the most frequent respiratory illnesses among admitted children. The highest admission rates occurred in children under one year of age, with a gradual decrease throughout childhood.

Correlation analysis using Spearman's rank coefficient suggested weak positive correlations between PM2.5 and PM10 concentrations with hospital admissions for respiratory illnesses. Notably, the strongest correlation was observed between weekly NOx exposure and total admissions for all diseases after log transformation.

Machine learning models using Support Vector Regression (SVR) and Partial Least Squares (PLS) regression achieved limited predictive power (accuracy of 0.18 and 0.25, respectively) in estimating the number of admissions based on air pollutant concentrations.

5.1 Limitations

This study is subject to several limitations. First, the analysis is based on data from Sant Joan de Reus Hospital (HUSJR) obtained on May 31st, 2024. Data from Joan XXIII Hospital (HUJ23) was not available at the time of this project submission even though we obtained the CEIm approval at the end of March. This limits the generalizability of the findings to the broader pediatric population in the Camp de Tarragona region.

Second, the analysis solely considered air pollutant concentrations measured at monitoring stations. Ideally, individual exposure levels for each patient would be incorporated. However, such data was not available for this study.

Third, weather conditions such as temperature, pressure, and wind speed were not included in the analysis. These factors have been shown to influence hospital admissions for respiratory illnesses[29] and should be considered in future studies.

Fourth, limitations exist in the spatial resolution of the air quality data. Many towns lack monitoring stations, needing the assignment of the closest station's data to patients' home cities within an arbitrary 10-kilometer radius. This approach reduces the accuracy

of exposure estimates and consequently reduces the size of the analyzed admissions dataset.

Fifth, not all pollutants are measured at every monitoring station. The analysis was restricted to PM2.5, PM10, SO_2, and NO_x, as these were the most consistently measured pollutants across the network. The inclusion of additional pollutants might provide further insights into the relationship between air quality and pediatric respiratory health.

These limitations highlight the need for future research to incorporate individual exposure data, consider weather conditions, and use a denser network of air quality monitoring stations. Additionally, future studies could explore the health effects of a wider range of pollutants.

5.2 Future perspectives

The observed correlations between air pollution and pediatric respiratory admissions suggest a potential public health concern in the Camp de Tarragona region. Further investigation is needed to confirm these associations and explore underlying mechanisms.

The findings highlight the potential value of incorporating air quality data into pediatric healthcare practices. Future research should explore individual exposure assessment methods to account for spatial variations and microenvironments as early identification of children at risk due to air pollution exposure could inform preventative measures.

Within the OnBREATHE project, future research endeavors will focus on expanding the scope of the current study. We aim to acquire pediatric respiratory health data from all healthcare centers across Catalonia. This broader dataset can be obtained through a formal request to the Health Quality and Assessment Agency of Catalonia (AQuAS) via their Data analytics program for health research and innovation (PADRIS). This comprehensive data would enhance the generalizability of our findings by encompassing a larger and more representative sample of the pediatric population in Catalonia.

Furthermore, future studies will incorporate weather data into the analysis. Including factors like temperature, pressure, and wind speed will allow for a more comprehensive assessment of environmental influences on pediatric respiratory health. These additional variables might strengthen the observed correlations between air quality and hospital admissions.

Finally, based on the strength of the correlations identified in this study, future research will explore the potential for incorporating multiple pollutants into machine learning models. By including a wider range of relevant air pollutants, these models could potentially achieve higher accuracy in predicting pediatric hospital admissions for respiratory illnesses.

References

[1] A. Zanobetti *et al.*, "Early-Life Exposure to Air Pollution and Childhood Asthma Cumulative Incidence in the ECHO CREW Consortium," *JAMA Netw Open*, vol. 7, no. 2, p. e240535, Feb. 2024, doi: 10.1001/jamanetworkopen.2024.0535.

[2] F. A. Wichmann *et al.*, "Increased asthma and respiratory symptoms in children exposed to petrochemical pollution," *Journal of Allergy and Clinical Immunology*, vol. 123, no. 3, pp. 632–638, Mar. 2009, doi: 10.1016/j.jaci.2008.09.052.

[3] World Health Organization: WHO, "Air pollution." Accessed: Apr. 05, 2024. [Online]. Available: https://www.who.int/health-topics/air-pollution

[4] A. Faustini *et al.*, "Air pollution and multiple acute respiratory outcomes," *European Respiratory Journal*, vol. 42, no. 2, pp. 304–313, Aug. 2013, doi: 10.1183/09031936.00128712.

[5] US EPA, "Nitrogen Oxides (NOx) Control Regulations." Accessed: May 14, 2024. [Online]. Available: https://www3.epa.gov/region1/airquality/nox.html

[6] US EPA, "Sulfur Dioxide Basics." Accessed: May 14, 2024. [Online]. Available: https://www.epa.gov/so2-pollution/sulfur-dioxide-basics

[7] World Health Organization, *WHO Guidelines for Indoor Air Quality: Selected Pollutants*. World Health Organization. Regional Office for Europe., 2010.

[8] California Air Resources Board, "Hydrogen sulfide & health." Accessed: May 14, 2024. [Online]. Available: https://ww2.arb.ca.gov/es/resources/hydrogen-sulfide-and-health

[9] US EPA, "Basic Information about Carbon Monoxide (CO) Outdoor Air Pollution." Accessed: May 14, 2024. [Online]. Available: https://www.epa.gov/co-pollution/basic-information-about-carbon-monoxide-co-outdoor-air-pollution

[10] American Lung Association, "Ozone." Accessed: May 14, 2024. [Online]. Available: https://www.lung.org/clean-air/outdoors/what-makes-air-unhealthy/ozone

[11] American Lung Association, "Particle pollution," Oct. 2023, Accessed: May 14, 2024. [Online]. Available: https://www.lung.org/clean-air/outdoors/what-makes-air-unhealthy/particle-pollution

[12] Generalitat de Catalunya. Departament d'Acció Climàtica; Alimentació i Agenda Rural, "Punts de mesurament i equipament de la Xarxa de Vigilància i Previsió de la Contaminació Atmosfèrica." Jan. 01, 2024.

[13] I. Hertz-Picciotto *et al.*, "Early Childhood Lower Respiratory Illness and Air Pollution," *Environ Health Perspect*, vol. 115, no. 10, pp. 1510–1518, Oct. 2007, doi: 10.1289/ehp.9617.

[14] A. Dondi *et al.*, “Outdoor Air Pollution and Childhood Respiratory Disease: The Role of Oxidative Stress,” *Int J Mol Sci*, vol. 24, no. 5, p. 4345, Feb. 2023, doi: 10.3390/ijms24054345.

[15] World Health Organization (WHO), “International Statistical Classification of Diseases and Related Health Problems (ICD).” Accessed: May 20, 2024. [Online]. Available: https://www.who.int/standards/classifications/classification-of-diseases

[16] C. J. Gross, J. J. Porter, S. C. Lipsett, M. C. Monuteaux, A. W. Hirsch, and M. I. Neuman, “Variation in Management and Outcomes of Children With Complicated Pneumonia,” *Hosp Pediatr*, vol. 11, no. 3, pp. 207–214, Mar. 2021, doi: 10.1542/hpeds.2020-001800.

[17] J. D. Baghdadi *et al.*, “Antibiotic Use and Bacterial Infection among Inpatients in the First Wave of COVID-19: a Retrospective Cohort Study of 64,691 Patients,” *Antimicrob Agents Chemother*, vol. 65, no. 11, Oct. 2021, doi: 10.1128/AAC.01341-21.

[18] US EPA, “Basic Information about NO2.” Accessed: May 20, 2024. [Online]. Available: https://www.epa.gov/no2-pollution/basic-information-about-no2

[19] American Lung Association, “Sulfur dioxide.” Accessed: May 20, 2024. [Online]. Available: https://www.lung.org/clean-air/outdoors/what-makes-air-unhealthy/sulfur-dioxide

[20] Environmental Health, “Benzene.” Accessed: May 20, 2024. [Online]. Available: https://www.vdh.virginia.gov/environmental-health/public-health-toxicology/benzene/

[21] Virginia Department of Health, “Hydrogen sulfide.” Accessed: May 20, 2024. [Online]. Available: https://www.vdh.virginia.gov/environmental-health/public-health-toxicology/hydrogen-sulfide/

[22] US EPA, “Basic Information about Carbon Monoxide (CO) Outdoor Air Pollution.” Accessed: May 20, 2024. [Online]. Available: https://www.epa.gov/co-pollution/basic-information-about-carbon-monoxide-co-outdoor-air-pollution

[23] US EPA, “Ground-level Ozone Basics.” Accessed: May 20, 2024. [Online]. Available: https://www.epa.gov/ground-level-ozone-pollution/ground-level-ozone-basics

[24] California Air Resources Board, “Inhalable Particulate Matter and Health (PM2.5 and PM10).” Accessed: May 20, 2024. [Online]. Available: https://ww2.arb.ca.gov/es/resources/inhalable-particulate-matter-and-health

[25] D. Austin, “2023 Internet Minute Infographic, by eDiscovery Today and LTMG!” Accessed: May 21, 2024. [Online]. Available: https://ediscoverytoday.com/2023/04/20/2023-internet-minute-infographic-by-ediscovery-today-and-ltmg-ediscovery-trends/

[26] Google, “Google Colaboratory: Frequently asked questions.” Accessed: May 21, 2024. [Online]. Available: https://research.google.com/colaboratory/intl/en-GB/faq.html

[27] I. Stancin and A. Jovic, “An overview and comparison of free Python libraries for data mining and big data analysis,” in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, May 2019, pp. 977–982. doi: 10.23919/MIPRO.2019.8757088.

[28] Medi Ambient i Sostenibilitat, “L’ Índex Català de Qualitat de l’Aire (ICQA).”

[29] J. Yang *et al.*, “Application of machine learning to predict hospital visits for respiratory diseases using meteorological and air pollution factors in Linyi, China,” *Environmental Science and Pollution Research*, vol. 30, no. 38, pp. 88431–88443, Jul. 2023, doi: 10.1007/s11356-023-28682-8.

Appendixes

Appendix A. CEIm approval

IISPV
INSTITUT D'INVESTIGACIÓ SANITÀRIA PERE VIRGILI

CEIm
Comitè Ètic d'Investigació amb medicaments

DICTAMEN COMITÉ ÉTICO DE INVESTIGACIÓN CON MEDICAMENTOS

FRANCESC XAVIER SUREDA BATLLE, Secretario del Comité Ético de Investigación con Medicamentos del IISPV da fe de los acuerdos aprobados con el visto bueno de JOSEP MARIA ALEGRET COLOMÉ que preside la reunión.

Este Comité, en su reunión de fecha **21/03/2024** acta número **003/2024** se ha evaluado y decidido emitir **Informe Favorable** para que se realice el estudio titulado:

"Estudi de la relació entre la contaminació atmosfèrica i salut pediàtrica al Camp de Tarragona"

Código: CASE_TGN
Versión Protocolo: Versió 1. 23/02/2024
Versión H.I.P. y Consentimiento Informado: Exención
Promotor: INVESTIGADOR
Ref. CEIM: 093/2024

CONSIDERA QUE:

- Se cumplen los requisitos necesarios de idoneidad del protocolo en relación con los objetivos del estudio y están justificados los riesgos y molestias previsibles para el sujeto.
- La capacidad del investigador y los medios disponibles son apropiados para lleva a cabo el estudio.
- Se acepta la exención de consentimiento propuesta para este estudio.
- El alcance de las compensaciones económicas previstas no interfiera con el respeto a los postulados éticos.

Este comité **acepta** que dicho estudio sea realizado en:

Institut d'Investigació Sanitaria Pere Virgili - IISPV por RAMÍREZ GONZÁLEZ, NOELIA del Servicio de Investigación en Pediatría, Nutrición y Desarrollo Humano
Hospital Universitari Sant Joan de Reus por ESCRIBANO SUBÍAS, JOAQUÍN del Servicio de Pediatría
Hospital Universitari Joan XXIII de Tarragona por VASQUEZ PÉREZ, AMALUI VANECSA del Servicio de Pediatría

En el caso que se evalúe algún proyecto en el que participe como investigador/colaborador algún miembro de este comité, se ausentará de la reunión durante la discusión del estudio.

La composición actual del CEIm del Instituto d'Investigació Sanitària Pere Virigili es la siguiente:

Presidente

Dr. Josep Mª Alegret Colomé
Cardiólogo. *Salut Sant Joan de Reus-Baix Camp.*

Vicepresidente

Dra. Maria Teresa Auguet Quintilla

Registre de Fundacions de la Generalitat de Catalunya núm. inscripció 2.206 – NIF G43814045

1 / 2

Generalitat de Catalunya · S/ICS · S/ HJ23 Hospital Universitari Joan XXIII ICS Camp de Tarragona · S/ HTVC Hospital de Tortosa Verge de la Cinta · Salut/ Salut Sant Joan Reus - Baix Camp · GRUP PERE MATA · UNIVERSITAT ROVIRA I VIRGILI

www.iispv.cat

Servicio de Medicina Interna. Hospital Universitari Joan XXIII. Representante de la Comisión de Investigación.

Secretario

Dr. Francesc Xavier Sureda Batlle
Profesor Titular de Farmacología. Universitat Rovira i Virgili.

Vocales

Sra. Mònica Cots Morenilla
Unidad de Atención Usuario. Hospital Universitari Joan XXIII.
Dr. Joaquín Escribano Súbias.
Médico del Servicio de Pediatría. Representante de la Comisión de Bioètica Assitencial. Salut Sant Joan de Reus-Baix Camp.
Dra. Gemma Flores Mateo
Servicio de Medicina Preventiva y Salud Pública. Xarxa Santiaria Santa Tecla
Sra. Elisabet Galve Aixa
Delegada en Protección de Datos del IISPV
Sra. M. Mar Granell Barceló
Abogada i Asesora Jurídica del Comité.
Dra. M. Francisca Jiménez Herrera
Profesora Titular Universitaria Departamento Enfermería. Universidad Rovira i Virgili
Dr. Jesús Miguel López-Dupla
Servicio de Medicina Interna Hospital Universitari Joan XXIII
Dr. Jordi Mallol Mirón
Catedrático de Farmacología.
Dr. Donis Mas Rosell
Medicina Psiquiatria - Institut Pere Mata.
Dra. Montserrat Olona Cabezas
Medicina Preventiva i Epidemiologia. *Hospital Universitari Joan XXIII*
Dra. Mª Angels Roch Ventura
Farmacia Hospitalaria Hospital Universitari Joan XXIII
Sra. Isabel Rosich Martí
Farmacéutica Atención Primaria
Dr. Xavier Ruiz Plazas
Urología. Hospital Universitari Joan XXIII.
Sra. Meritxell Torres Paisal
Delegada Protección de Datos - IISPV
Sra. Mercè Vilella Papaseit
Representante de la Sociedad Civil

Firma Francesc Xavier Sureda Batlle - DNI 38088115T (TCAT)
Firmado digitalmente por Francesc Xavier Sureda Batlle - DNI 38088115T (TCAT)
Fecha: 2024.03.22 15:16:08 +01'00'

Dr. Francesc Xavier Sureda
Secretario CEIm IISPV

Appendix B. Air quality measurements grouped by monitoring station, pollutant and year

Station	Pollutant	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
Alcover	CO	0.22 ± 0.07	0.23 ± 0.07	0.25 ± 0.1	0.25 ± 0.12	0.23 ± 0.09	0.25 ± 0.09	0.27 ± 0.16	0.23 ± 0.09	0.25 ± 0.08	0.24 ± 0.1	0.32 ± 0.17
	H2S	1.05 ± 0.24	1.07 ± 0.81	1.19 ± 0.95	1.29 ± 1.06	2.04 ± 1.6	1.73 ± 1.22	2.26 ± 1.51	2.13 ± 2.07	1.51 ± 0.73	1.3 ± 0.33	1.59 ± 0.78
	NOX	14.48 ± 15.34	15.56 ± 17.52	17.64 ± 20.14	14.23 ± 16.76	15.23 ± 16.89	13.52 ± 16.98	13.63 ± 13.76	12.34 ± 17.68	15.96 ± 16.24	13.23 ± 12.28	11.49 ± 9.45
	O3	73.03 ± 22.15	69.15 ± 23.06	69.86 ± 24.89	69.21 ± 21.66	71.73 ± 21.54	67.93 ± 23.32	72.54 ± 24.97	65.73 ± 22.55	65.73 ± 20.34	71.36 ± 23.19	71.16 ± 21.39
Constantí	C6H6	1.96 ± 3.81	1.52 ± 2.64	1.65 ± 3.63	1.21 ± 1.79	1.58 ± 2.38	1.82 ± 2.79	1.38 ± 1.7	1.79 ± 6.85	0.93 ± 1.04	1.49 ± 3.43	1.69 ± 3.16
	H2S	1.75 ± 1.32	1.7 ± 1.21	1.94 ± 1.43	1.84 ± 1.69	1.9 ± 1.39	2.06 ± 2.25	1.4 ± 0.9	1.61 ± 4.93	1.39 ± 0.69	1.63 ± 1.3	2.08 ± 1.48
	NOX	19.0 ± 20.21	21.6 ± 20.03	24.44 ± 22.81	22.67 ± 19.95	21.9 ± 19.95	20.77 ± 19.88	22.99 ± 18.47	18.43 ± 19.15	18.1 ± 16.27	18.89 ± 16.32	16.94 ± 14.62
	O3	57.79 ± 29.32	55.69 ± 30.36	57.58 ± 32.11	55.76 ± 28.46	56.32 ± 29.07	57.32 ± 31.9	53.86 ± 30.67	53.77 ± 28.96	57.19 ± 27.37	57.17 ± 30.15	58.93 ± 28.48
	SO2	1.76 ± 3.41	1.89 ± 2.46	5.05 ± 4.91	5.1 ± 5.45	4.6 ± 5.14	5.38 ± 5.09	4.91 ± 10.26	3.54 ± 3.4	4.16 ± 4.02	3.21 ± 3.85	4.47 ± 4.37
Constantí (Gaudí)	PM10	19.38 ± 7.74	18.44 ± 10.76	23.28 ± 8.56	19.56 ± 8.85	19.87 ± 7.95	21.53 ± 9.47	19.87 ± 10.26	18.17 ± 10.7	17.09 ± 7.28	20.69 ± 7.58	17.4 ± 7.77
	PM2.5	12.33 ± 5.68	11.32 ± 6.69	14.71 ± 6.78	11.3 ± 5.34	11.13 ± 5.82	10.34 ± 4.67	10.11 ± 5.47	9.01 ± 4.89	8.54 ± 4.21	9.11 ± 4.45	9.05 ± 5.1
El Morell (Deixalleria municipal)	C6H6			1.09 ± 0.22	1.12 ± 0.3	2.76 ± 3.28	1.99 ± 1.4	1.0 ± 1.46	0.34 ± 0.45	0.96 ± 0.79	1.17 ± 1.12	3.61 ± 4.54
La Canonja (deixalleria municipal)	C6H6					2.9 ± 1.93	1.67 ± 0.83	0.85 ± 1.68	0.22 ± 0.21	0.53 ± 0.46	1.48 ± 1.22	3.18 ± 3.57
Perafort (Puigdelfí)	C6H6	0.77 ± 0.94	0.71 ± 0.48	0.74 ± 0.61	0.71 ± 0.68	0.74 ± 0.64	0.87 ± 1.28	0.74 ± 0.62	0.79 ± 0.7	0.72 ± 0.55	0.71 ± 0.62	1.02 ± 1.33
	H2S	1.29 ± 0.67	1.32 ± 0.71	1.49 ± 0.72	1.48 ± 0.69	1.44 ± 0.48	1.4 ± 0.49	1.49 ± 0.59	1.72 ± 0.7	1.86 ± 0.64	1.72 ± 0.51	1.52 ± 0.4
	NOX	15.63 ± 11.39	15.03 ± 12.18	15.02 ± 13.65	13.99 ± 10.22	13.04 ± 9.72	11.27 ± 8.58	12.02 ± 8.21	12.7 ± 8.77	14.14 ± 9.95	11.76 ± 8.17	11.84 ± 7.56
	PM10									14.76 ± 9.41	16.05 ± 9.53	15.08 ± 10.24
	PM2.5									9.76 ± 7.3	10.19 ± 6.76	10.1 ± 7.21
	SO2	2.71 ± 3.45	1.76 ± 2.36	1.96 ± 2.55	2.49 ± 5.01	2.57 ± 3.56	2.13 ± 3.22	2.42 ± 4.49	1.62 ± 2.44	1.79 ± 2.67	1.98 ± 3.07	2.37 ± 3.57
Reus	CO	0.3 ± 0.14	0.23 ± 0.09	0.26 ± 0.19	0.26 ± 0.11	0.25 ± 0.12	0.23 ± 0.1	0.23 ± 0.09	0.24 ± 0.09	0.28 ± 0.12	0.28 ± 0.13	0.26 ± 0.11
	H2S	1.12 ± 0.28	1.45 ± 0.67	1.25 ± 0.46	1.13 ± 0.47	1.16 ± 0.37	1.15 ± 0.36	1.17 ± 0.31	1.22 ± 0.29	1.95 ± 1.16	1.64 ± 0.58	1.94 ± 0.91

	NOX	29.63 ± 40.92	28.03 ± 33.91	31.2 ± 45.52	31.59 ± 43.22	29.04 ± 38.39	26.88 ± 35.61	26.97 ± 34.82	23.82 ± 33.69	23.52 ± 31.45	26.79 ± 33.88	22.75 ± 25.82
	O3	63.01 ± 28.75	58.86 ± 29.13	58.22 ± 29.83	56.46 ± 25.62	57.67 ± 28.73	61.43 ± 30.33	61.92 ± 30.06	54.99 ± 26.96	60.54 ± 26.32	58.55 ± 30.19	57.92 ± 27.2
	PM10	20.34 ± 14.72	23.79 ± 18.17	25.01 ± 17.49	22.94 ± 18.22	22.55 ± 14.47	19.38 ± 12.98	17.68 ± 12.86	18.53 ± 13.86	20.75 ± 14.97	22.97 ± 14.05	19.7 ± 13.74
Reus (el Tallapedra)	C6H6	0.78 ± 0.52	1.07 ± 0.2	1.14 ± 0.32	1.05 ± 0.17	1.04 ± 0.17	0.79 ± 0.47	0.73 ± 0.41	0.79 ± 0.68	0.74 ± 0.35	0.8 ± 0.47	0.77 ± 0.6
Tarragona (Bonavista)	H2S	1.57 ± 0.8	1.89 ± 0.88	1.77 ± 1.1	1.81 ± 0.94	1.64 ± 0.91	1.84 ± 1.37	1.43 ± 0.74	1.26 ± 0.48	1.32 ± 0.59	1.54 ± 0.87	1.87 ± 1.1
	NOX	25.38 ± 27.2	28.71 ± 30.13	33.72 ± 33.17	29.45 ± 28.46	29.49 ± 25.52	27.54 ± 28.51	23.44 ± 23.78	20.96 ± 22.17	19.19 ± 21.18	23.91 ± 23.8	19.1 ± 19.3
	PM10	15.41 ± 10.2	16.56 ± 11.04	22.42 ± 12.76	17.19 ± 11.12	19.16 ± 11.56	19.24 ± 13.17	19.4 ± 12.86	15.26 ± 12.6	21.11 ± 14.34	23.17 ± 14.97	19.45 ± 14.87
	PM2.5	9.73 ± 7.65	10.99 ± 8.11	13.54 ± 8.99	10.49 ± 7.15	11.6 ± 7.97	11.16 ± 7.19	10.53 ± 7.65	9.45 ± 8.09	11.44 ± 8.22	11.89 ± 7.66	9.94 ± 6.93
	SO2	1.48 ± 2.09	3.25 ± 2.92	3.08 ± 2.33	3.11 ± 2.61	3.37 ± 2.7	2.74 ± 2.59	3.31 ± 3.04	2.43 ± 2.54	3.21 ± 2.69	3.44 ± 3.16	2.88 ± 3.06
Tarragona (Parc de la Ciutat)	CO	0.3 ± 0.15	0.25 ± 0.11	0.24 ± 0.09	0.27 ± 0.09	0.23 ± 0.08	0.24 ± 0.1	0.25 ± 0.1	0.22 ± 0.06	0.21 ± 0.04	0.22 ± 0.08	0.26 ± 0.1
	H2S	1.61 ± 0.67	1.59 ± 0.62	1.99 ± 0.97	1.9 ± 0.84	1.98 ± 0.91	1.99 ± 0.96	2.67 ± 1.4	2.43 ± 1.5	1.25 ± 0.56	1.45 ± 1.09	2.13 ± 0.88
	NOX	35.92 ± 46.76	34.59 ± 45.0	37.53 ± 48.87	32.66 ± 39.98	33.92 ± 44.65	31.48 ± 41.81	29.53 ± 37.05	24.94 ± 36.76	23.66 ± 30.22	28.22 ± 37.8	26.51 ± 34.32
	O3	56.3 ± 31.27	52.37 ± 32.1	52.95 ± 32.76	52.22 ± 30.48	54.9 ± 31.76	55.83 ± 32.23	58.08 ± 32.0	56.47 ± 29.65	60.16 ± 29.27	59.28 ± 33.88	59.54 ± 30.3
	SO2	2.47 ± 2.17	2.08 ± 1.23	2.02 ± 1.81	1.78 ± 2.38	1.72 ± 2.21	1.66 ± 1.94	1.7 ± 2.2	1.36 ± 1.16	1.4 ± 1.22	1.56 ± 2.16	2.46 ± 2.41
	C6H6	0.92 ± 0.78	1.05 ± 0.14	1.11 ± 0.34	1.1 ± 0.66	1.13 ± 0.31	0.78 ± 0.28	0.86 ± 0.78	0.9 ± 0.7	0.9 ± 0.7	0.88 ± 0.47	0.87 ± 0.56
Tarragona (Salut)	PM10				20.82 ± 9.42	22.29 ± 7.82	20.31 ± 7.51	21.8 ± 12.23	17.98 ± 10.6	19.12 ± 7.71	20.36 ± 7.32	19.74 ± 9.2
	PM2.5											9.92 ± 5.33
Tarragona (Sant Salvador)	H2S	1.76 ± 1.74	1.24 ± 0.55	1.26 ± 0.59	1.17 ± 0.54	1.31 ± 0.54	1.27 ± 0.54	1.28 ± 0.36	1.23 ± 0.55	1.22 ± 0.31	1.27 ± 0.45	1.42 ± 0.39
	NOX	32.14 ± 28.17	28.74 ± 25.84	30.98 ± 28.24	27.29 ± 24.02	26.14 ± 22.94	23.64 ± 22.37	24.44 ± 21.67	20.1 ± 20.63	21.46 ± 20.28	22.46 ± 20.24	19.63 ± 14.12
	SO2	2.86 ± 5.06	1.76 ± 2.29	1.84 ± 2.19	1.82 ± 3.45	3.21 ± 4.98	2.43 ± 4.04	2.98 ± 4.67	1.96 ± 3.88	1.98 ± 3.61	2.11 ± 3.82	3.14 ± 5.09
Tarragona (Universitat Laboral)	H2S	1.49 ± 1.81	1.47 ± 1.36	1.65 ± 1.77	1.86 ± 2.58	1.98 ± 2.61	1.54 ± 2.67	2.62 ± 2.83	2.11 ± 2.43	1.54 ± 1.29	2.64 ± 2.7	3.25 ± 2.74
	NOX	26.5 ± 28.84	28.62 ± 31.99	28.73 ± 32.89	29.13 ± 31.22	30.64 ± 38.55	27.17 ± 34.63	25.35 ± 26.89	20.86 ± 25.09	20.85 ± 22.87	25.38 ± 26.88	19.58 ± 20.81
	PM10	16.57 ± 6.34	19.51 ± 9.99	20.82 ± 9.39	19.79 ± 9.59	21.18 ± 7.64	19.14 ± 7.3	21.09 ± 11.33	17.21 ± 8.92	17.17 ± 7.42	19.56 ± 8.47	16.57 ± 6.77
	SO2	2.53 ± 2.08	2.51 ± 2.31	2.27 ± 2.12	1.9 ± 3.01	2.26 ± 3.37	1.53 ± 1.71	1.53 ± 1.92	1.11 ± 0.99	1.14 ± 0.74	1.25 ± 1.5	1.48 ± 1.93

Vandellòs i l'Hospitalet de l'Infant (viver)	NOX			14.36 ± 13.24	13.1 ± 12.43	13.4 ± 13.08	12.09 ± 10.87	10.01 ± 8.13	8.54 ± 11.62	14.22 ± 16.58	12.38 ± 14.54	12.38 ± 13.2
	PM10	10.93 ± 5.39	15.06 ± 9.58	16.32 ± 9.8	13.4 ± 12.56	13.89 ± 7.84	13.13 ± 7.76	17.06 ± 15.15	13.69 ± 9.06	14.23 ± 8.71	15.06 ± 8.27	13.69 ± 8.62
	SO2	1.27 ± 0.71	1.32 ± 1.19	1.59 ± 1.09	1.59 ± 0.94	1.38 ± 0.71	1.27 ± 0.87	1.26 ± 0.87	1.17 ± 0.68	1.13 ± 0.63	1.46 ± 1.33	1.47 ± 0.81
Vila-seca (IES Vila-seca)	C6H6								0.8 ± 0.51	0.77 ± 0.44	0.91 ± 0.46	0.59 ± 0.21
	H2S								1.06 ± 0.14	1.19 ± 0.24	1.22 ± 0.31	1.23 ± 0.38
	NOX								22.57 ± 32.32	18.82 ± 24.74	23.5 ± 31.62	18.59 ± 25.75
	O3								48.95 ± 26.27	62.4 ± 27.55	58.5 ± 31.16	61.41 ± 28.29
	PM10								13.23 ± 11.11	17.87 ± 12.88	17.98 ± 15.49	17.15 ± 10.36
	PM2.5								9.3 ± 9.37	11.65 ± 9.97	11.59 ± 9.37	8.31 ± 5.81
	SO2								1.87 ± 1.6	2.52 ± 1.81	2.76 ± 2.51	2.7 ± 2.42

Appendix C. Classification of days based on ICQA levels per station, pollutant and year

Station	Air pollutant	Year	Good	Reasonably good	Regular	Unfavourable	Very unfavourable
Alcover	CO	2013	360	0	0	0	0
	H2S	2013	350	0	0	0	0
	NOX	2013	355	1	0	0	0
	SO2	2013	360	0	0	0	0
	O3	2013	29	317	14	0	0
	H2S	2014	353	0	0	0	0
	SO2	2014	351	0	0	0	0
	O3	2014	50	289	12	0	0
	CO	2014	344	0	0	0	0
	NOX	2014	353	7	0	0	0
	O3	2015	59	284	18	0	0
	SO2	2015	360	0	0	0	0
	CO	2015	361	0	0	0	0
	H2S	2015	351	0	0	0	0
	NOX	2015	354	6	0	0	0
	H2S	2016	346	0	0	0	0
	SO2	2016	363	0	0	0	0
	CO	2016	366	0	0	0	0
	NOX	2016	363	3	0	0	0
	O3	2016	48	312	5	1	0
	H2S	2017	329	0	0	0	0
	O3	2017	26	330	9	0	0
	NOX	2017	350	4	0	0	0
	CO	2017	341	0	0	0	0

SO2	2017	365	0	0	0	0
H2S	2018	365	0	0	0	0
CO	2018	361	0	0	0	0
NOX	2018	361	3	0	0	0
O3	2018	52	295	6	0	0
SO2	2018	365	0	0	0	0
NOX	2019	365	0	0	0	0
H2S	2019	363	0	0	0	0
SO2	2019	365	0	0	0	0
O3	2019	53	292	20	0	0
CO	2019	365	0	0	0	0
CO	2020	361	0	0	0	0
O3	2020	56	303	3	0	0
H2S	2020	350	0	0	0	0
SO2	2020	361	0	0	0	0
NOX	2020	357	2	0	0	0
CO	2021	364	0	0	0	0
NOX	2021	350	9	0	0	0
SO2	2021	361	0	0	0	0
H2S	2021	365	0	0	0	0
O3	2021	36	327	2	0	0
CO	2022	365	0	0	0	0
H2S	2022	365	0	0	0	0
SO2	2022	358	0	0	0	0
NOX	2022	363	0	0	0	0
O3	2022	37	314	9	0	0
H2S	2023	352	0	0	0	0
CO	2023	332	0	0	0	0

	NOX	2023	365	0	0	0	0
	O3	2023	24	334	7	0	0
	SO2	2023	365	0	0	0	0
	NOX	2013	317	24	0	0	0
	H2S	2013	363	0	0	0	0
	SO2	2013	364	0	0	0	0
	C6H6	2013	302	11	0	0	0
	O3	2013	126	231	2	0	0
	H2S	2014	365	0	0	0	0
	SO2	2014	265	0	0	0	0
	NOX	2014	334	29	0	0	0
	O3	2014	138	227	0	0	0
	C6H6	2014	325	5	0	0	0
	H2S	2015	356	0	0	0	0
	O3	2015	127	232	4	0	0
Constantí	NOX	2015	317	45	2	0	0
	SO2	2015	364	0	0	0	0
	C6H6	2015	337	5	2	0	0
	C6H6	2016	351	2	0	0	0
	NOX	2016	335	31	0	0	0
	O3	2016	125	241	0	0	0
	H2S	2016	366	0	0	0	0
	SO2	2016	366	0	0	0	0
	SO2	2017	365	0	0	0	0
	NOX	2017	333	32	0	0	0
	C6H6	2017	362	3	0	0	0
	H2S	2017	365	0	0	0	0
	O3	2017	124	240	0	0	0

NOX	2018	337	25	0	0	0
O3	2018	129	234	1	0	0
C6H6	2018	348	1	0	0	0
SO2	2018	354	0	0	0	0
H2S	2018	364	0	0	0	0
O3	2019	147	216	2	0	0
C6H6	2019	365	0	0	0	0
H2S	2019	364	0	0	0	0
NOX	2019	341	23	0	0	0
SO2	2019	363	0	0	0	0
NOX	2020	342	24	0	0	0
C6H6	2020	348	12	2	1	0
O3	2020	135	231	0	0	0
H2S	2020	366	0	0	0	0
SO2	2020	357	0	0	0	0
H2S	2021	364	0	0	0	0
SO2	2021	352	0	0	0	0
O3	2021	117	248	0	0	0
NOX	2021	348	14	0	0	0
C6H6	2021	352	0	0	0	0
SO2	2022	352	0	0	0	0
O3	2022	133	232	0	0	0
C6H6	2022	361	2	0	1	0
NOX	2022	350	11	0	0	0
H2S	2022	365	0	0	0	0
SO2	2023	300	0	0	0	0
O3	2023	108	251	0	0	0
C6H6	2023	359	4	2	0	0

	H2S	2023	365	0	0	0	0
	NOX	2023	355	10	0	0	0
Constantí (Gaudí)	PM10	2013	108	68	1	1	0
	PM2.5	2013	76	91	7	4	0
	PM10	2014	122	45	3	3	0
	PM2.5	2014	98	74	4	5	0
	PM2.5	2015	52	99	16	10	0
	PM10	2015	75	98	1	3	0
	PM10	2016	110	69	1	1	0
	PM2.5	2016	91	76	6	3	0
	PM2.5	2017	105	58	11	4	0
	PM10	2017	102	88	1	0	0
	PM2.5	2018	96	59	7	0	0
	PM10	2018	88	68	3	2	0
	PM10	2019	116	50	4	3	0
	PM2.5	2019	113	52	3	4	0
	PM2.5	2020	127	41	4	1	0
	PM10	2020	130	39	1	3	0
	PM10	2021	217	53	3	0	0
	PM2.5	2021	228	41	4	1	0
	PM2.5	2022	234	64	7	2	0
	PM10	2022	173	142	3	0	0
	PM10	2023	221	70	4	1	0
	PM2.5	2023	233	59	4	6	0
El Morell (Deixalleria municipal)	C6H6	2015	73	0	0	0	0
	C6H6	2016	99	0	0	0	0
	C6H6	2017	43	3	0	1	0

	C6H6	2018	14	0	0	0	0
	C6H6	2019	95	3	0	0	0
	C6H6	2020	126	0	0	0	0
	C6H6	2021	75	0	0	0	0
	C6H6	2022	57	0	0	0	0
	C6H6	2023	36	6	5	0	0
La Canonja (deixalleria municipal)	C6H6	2017	30	2	0	0	0
	C6H6	2018	18	0	0	0	0
	C6H6	2019	105	0	1	0	0
	C6H6	2020	142	0	0	0	0
	C6H6	2021	78	0	0	0	0
	C6H6	2022	50	0	0	0	0
	C6H6	2023	40	9	1	0	0
Perafort (Puigdelfí)	NOX	2013	364	1	0	0	0
	SO2	2013	365	0	0	0	0
	C6H6	2013	297	0	0	0	0
	H2S	2013	362	0	0	0	0
	H2S	2014	364	0	0	0	0
	SO2	2014	365	0	0	0	0
	C6H6	2014	228	0	0	0	0
	NOX	2014	348	4	0	0	0
	C6H6	2015	353	0	0	0	0
	NOX	2015	358	7	0	0	0
	SO2	2015	365	0	0	0	0
	H2S	2015	365	0	0	0	0
	SO2	2016	364	0	0	0	0
	NOX	2016	363	1	0	0	0

C6H6	2016	352	0	0	0	0
H2S	2016	364	0	0	0	0
H2S	2017	365	0	0	0	0
SO2	2017	365	0	0	0	0
NOX	2017	365	0	0	0	0
C6H6	2017	361	0	0	0	0
C6H6	2018	362	0	0	0	0
H2S	2018	365	0	0	0	0
NOX	2018	358	0	0	0	0
SO2	2018	365	0	0	0	0
NOX	2019	363	0	0	0	0
SO2	2019	365	0	0	0	0
C6H6	2019	353	0	0	0	0
H2S	2019	362	0	0	0	0
NOX	2020	366	0	0	0	0
C6H6	2020	363	0	0	0	0
SO2	2020	366	0	0	0	0
H2S	2020	366	0	0	0	0
PM10	2021	101	21	0	0	0
NOX	2021	356	4	0	0	0
SO2	2021	363	0	0	0	0
H2S	2021	362	0	0	0	0
PM2.5	2021	80	34	6	2	0
C6H6	2021	362	0	0	0	0
SO2	2022	365	0	0	0	0
PM10	2022	294	70	1	0	0
H2S	2022	365	0	0	0	0
PM2.5	2022	227	122	12	4	0

	NOX	2022	360	1	0	0	0
	C6H6	2022	365	0	0	0	0
	PM2.5	2023	173	83	14	4	0
	SO2	2023	365	0	0	0	0
	NOX	2023	353	0	0	0	0
	H2S	2023	299	0	0	0	0
	PM10	2023	224	48	2	0	0
	C6H6	2023	363	1	0	0	0
Reus	CO	2013	364	0	0	0	0
	NOX	2013	276	52	9	5	0
	O3	2013	87	275	3	0	0
	H2S	2013	361	0	0	0	0
	PM10	2013	206	147	8	1	0
	O3	2014	108	252	2	0	0
	NOX	2014	241	64	3	1	0
	PM10	2014	161	171	17	7	1
	H2S	2014	355	0	0	0	0
	CO	2014	362	0	0	0	0
	H2S	2015	365	0	0	0	0
	NOX	2015	274	52	13	9	0
	CO	2015	365	0	0	0	0
	O3	2015	111	248	1	0	0
	PM10	2015	150	171	20	15	0
	PM10	2016	163	189	10	3	1
	NOX	2016	283	59	14	7	0
	H2S	2016	361	0	0	0	0
	CO	2016	366	0	0	0	0
	O3	2016	122	244	0	0	0

O3	2017	97	213	2	0	0
H2S	2017	361	0	0	0	0
PM10	2017	171	177	10	2	0
NOX	2017	288	53	11	3	0
CO	2017	351	0	0	0	0
NOX	2018	299	57	5	1	0
H2S	2018	362	0	0	0	0
O3	2018	106	252	2	0	0
CO	2018	359	0	0	0	0
PM10	2018	224	134	3	1	0
O3	2019	99	260	3	0	0
PM10	2019	245	94	1	3	0
CO	2019	358	0	0	0	0
H2S	2019	362	0	0	0	0
NOX	2019	290	62	3	0	0
PM10	2020	258	98	1	3	1
CO	2020	358	0	0	0	0
O3	2020	119	242	0	0	0
H2S	2020	361	0	0	0	0
NOX	2020	306	48	6	0	0
CO	2021	365	0	0	0	0
H2S	2021	360	0	0	0	0
PM10	2021	213	137	5	2	1
NOX	2021	296	38	4	3	0
O3	2021	81	281	2	0	0
CO	2022	350	0	0	0	0
O3	2022	114	250	1	0	0
H2S	2022	365	0	0	0	0

	PM10	2022	153	188	16	1	0
	NOX	2022	248	57	3	0	0
	PM10	2023	235	124	3	3	0
	NOX	2023	285	33	1	0	0
	H2S	2023	365	0	0	0	0
	O3	2023	112	253	0	0	0
	CO	2023	365	0	0	0	0
Reus (el Tallapedra)	C6H6	2013	104	0	0	0	0
	C6H6	2014	96	0	0	0	0
	C6H6	2015	117	0	0	0	0
	C6H6	2016	103	0	0	0	0
	C6H6	2017	56	0	0	0	0
	C6H6	2018	88	0	0	0	0
	C6H6	2019	65	0	0	0	0
	C6H6	2020	57	0	0	0	0
	C6H6	2021	82	0	0	0	0
	C6H6	2022	106	0	0	0	0
	C6H6	2023	109	0	0	0	0
Tarragona (Bonavista)	NOX	2013	312	50	3	0	0
	SO2	2013	365	0	0	0	0
	PM2.5	2013	232	109	13	8	0
	PM10	2013	279	82	1	0	0
	H2S	2013	363	0	0	0	0
	SO2	2014	359	0	0	0	0
	H2S	2014	349	0	0	0	0
	PM2.5	2014	214	121	14	11	0
	NOX	2014	268	78	2	0	0

PM10	2014	270	81	7	2	0
PM2.5	2015	139	143	31	20	0
SO2	2015	365	0	0	0	0
H2S	2015	361	0	0	0	0
NOX	2015	266	92	6	1	0
PM10	2015	168	151	11	6	0
PM10	2016	209	85	2	0	0
NOX	2016	255	72	2	0	0
SO2	2016	364	0	0	0	0
PM2.5	2016	179	100	10	5	0
H2S	2016	364	0	0	0	0
SO2	2017	359	0	0	0	0
PM2.5	2017	199	128	26	12	0
PM10	2017	235	124	6	0	0
NOX	2017	298	67	0	0	0
H2S	2017	362	0	0	0	0
SO2	2018	362	0	0	0	0
H2S	2018	359	0	0	0	0
PM2.5	2018	158	130	12	4	0
PM10	2018	183	115	5	1	0
NOX	2018	297	63	1	0	0
PM2.5	2019	221	118	14	10	0
H2S	2019	363	0	0	0	0
SO2	2019	363	0	0	0	0
NOX	2019	327	34	0	0	0
PM10	2019	227	124	10	2	0
NOX	2020	327	38	1	0	0
PM10	2020	291	68	3	3	0

	H2S	2020	366	0	0	0	0
	PM2.5	2020	259	81	17	8	0
	SO2	2020	365	0	0	0	0
	PM2.5	2021	213	117	17	18	0
	NOX	2021	320	25	0	0	0
	PM10	2021	188	163	9	5	0
	H2S	2021	360	0	0	0	0
	SO2	2021	363	0	0	0	0
	NOX	2022	327	36	1	0	0
	H2S	2022	364	0	0	0	0
	PM2.5	2022	183	141	24	10	0
	PM10	2022	153	176	16	6	0
	SO2	2022	364	0	0	0	0
	PM2.5	2023	219	112	9	7	0
	SO2	2023	344	0	0	0	0
	NOX	2023	338	20	0	0	0
	PM10	2023	230	104	10	3	0
	H2S	2023	357	0	0	0	0
Tarragona (Parc de la Ciutat)	H2S	2013	365	0	0	0	0
	C6H6	2013	97	1	0	0	0
	NOX	2013	266	82	13	4	0
	CO	2013	354	0	0	0	0
	SO2	2013	365	0	0	0	0
	O3	2013	135	230	0	0	0
	SO2	2014	362	0	0	0	0
	H2S	2014	326	0	0	0	0
	C6H6	2014	99	0	0	0	0
	CO	2014	325	0	0	0	0

NOX	2014	258	88	13	3	0
O3	2014	154	207	1	0	0
SO2	2015	365	0	0	0	0
NOX	2015	254	85	12	8	0
C6H6	2015	102	0	0	0	0
H2S	2015	365	0	0	0	0
CO	2015	363	0	0	0	0
O3	2015	147	216	1	0	0
O3	2016	154	211	0	0	0
CO	2016	366	0	0	0	0
C6H6	2016	78	0	0	0	0
SO2	2016	366	0	0	0	0
H2S	2016	363	0	0	0	0
NOX	2016	276	82	8	0	0
NOX	2017	274	78	7	4	0
SO2	2017	363	0	0	0	0
C6H6	2017	50	0	0	0	0
CO	2017	363	0	0	0	0
H2S	2017	352	0	0	0	0
O3	2017	127	230	0	0	0
NOX	2018	284	70	6	5	0
SO2	2018	365	0	0	0	0
C6H6	2018	86	0	0	0	0
O3	2018	143	205	3	0	0
CO	2018	365	0	0	0	0
H2S	2018	365	0	0	0	0
NOX	2019	282	77	6	0	0
SO2	2019	365	0	0	0	0

C6H6	2019	55	0	0	0	0
O3	2019	124	236	2	0	0
CO	2019	365	0	0	0	0
H2S	2019	365	0	0	0	0
H2S	2020	357	0	0	0	0
O3	2020	117	232	1	0	0
C6H6	2020	52	0	0	0	0
CO	2020	366	0	0	0	0
SO2	2020	363	0	0	0	0
NOX	2020	308	51	6	1	0
H2S	2021	362	0	0	0	0
O3	2021	98	255	3	0	0
C6H6	2021	75	0	0	0	0
NOX	2021	329	31	3	1	0
SO2	2021	364	0	0	0	0
CO	2021	363	0	0	0	0
O3	2022	118	238	5	0	0
CO	2022	351	0	0	0	0
C6H6	2022	107	0	0	0	0
NOX	2022	297	59	7	2	0
H2S	2022	365	0	0	0	0
SO2	2022	365	0	0	0	0
O3	2023	119	240	1	0	0
SO2	2023	365	0	0	0	0
C6H6	2023	102	0	0	0	0
NOX	2023	309	51	3	2	0
CO	2023	365	0	0	0	0
H2S	2023	365	0	0	0	0

Tarragona (Salut)	PM10	2016	109	62	4	1	0
	PM10	2017	98	100	4	0	0
	PM10	2018	102	79	3	0	0
	PM10	2019	114	66	2	6	1
	PM10	2020	157	44	4	3	0
	PM10	2021	135	54	4	1	0
	PM10	2022	122	68	2	1	0
	PM10	2023	136	57	4	4	0
	PM2.5	2023	204	69	6	7	0
Tarragona (Sant Salvador)	SO2	2013	365	0	0	0	0
	NOX	2013	279	83	1	0	0
	H2S	2013	355	0	0	0	0
	H2S	2014	365	0	0	0	0
	SO2	2014	365	0	0	0	0
	NOX	2014	302	62	1	0	0
	SO2	2015	365	0	0	0	0
	NOX	2015	292	73	0	0	0
	H2S	2015	365	0	0	0	0
	H2S	2016	365	0	0	0	0
	NOX	2016	308	54	1	0	0
	SO2	2016	362	0	0	0	0
	C6H6	2016	27	0	0	0	0
	NOX	2017	308	42	0	0	0
	H2S	2017	365	0	0	0	0
	SO2	2017	365	0	0	0	0
	C6H6	2017	52	0	0	0	0
	SO2	2018	362	0	0	0	0

	NOX	2018	321	31	0	0	0
	H2S	2018	362	0	0	0	0
	C6H6	2018	90	0	0	0	0
	SO2	2019	365	0	0	0	0
	H2S	2019	363	0	0	0	0
	NOX	2019	336	29	0	0	0
	C6H6	2019	65	0	0	0	0
	NOX	2020	335	31	0	0	0
	SO2	2020	365	0	0	0	0
	H2S	2020	365	0	0	0	0
	C6H6	2020	56	0	0	0	0
	SO2	2021	363	0	0	0	0
	H2S	2021	363	0	0	0	0
	NOX	2021	311	28	0	0	0
	C6H6	2021	75	0	0	0	0
	NOX	2022	333	32	0	0	0
	SO2	2022	365	0	0	0	0
	H2S	2022	365	0	0	0	0
	C6H6	2022	110	0	0	0	0
	H2S	2023	365	0	0	0	0
	NOX	2023	352	12	0	0	0
	SO2	2023	365	0	0	0	0
	C6H6	2023	109	0	0	0	0
Tarragona (Universitat Laboral)	NOX	2013	309	54	2	0	0
	SO2	2013	365	0	0	0	0
	H2S	2013	365	0	0	0	0
	PM2.5	2013	87	90	4	1	0
	C6H6	2013	105	0	0	0	0

PM10	2013	143	41	0	0	0
SO2	2014	365	0	0	0	0
NOX	2014	279	73	2	0	0
H2S	2014	365	0	0	0	0
PM2.5	2014	103	62	7	7	0
C6H6	2014	104	0	1	0	0
PM10	2014	131	46	1	6	0
SO2	2015	365	0	0	0	0
NOX	2015	286	69	5	1	0
H2S	2015	365	0	0	0	0
PM2.5	2015	49	105	11	11	0
C6H6	2015	113	1	1	0	0
PM10	2015	128	61	5	4	0
H2S	2016	365	0	0	0	0
NOX	2016	300	65	0	0	0
SO2	2016	365	0	0	0	0
PM10	2016	120	68	3	1	0
C6H6	2016	102	1	0	0	0
PM2.5	2016	97	71	7	4	0
NOX	2017	283	70	4	2	0
SO2	2017	363	0	0	0	0
H2S	2017	361	0	0	0	0
PM2.5	2017	91	75	11	3	0
C6H6	2017	51	1	0	0	0
PM10	2017	96	88	3	0	0
NOX	2018	305	52	4	1	0
H2S	2018	365	0	0	0	0
SO2	2018	365	0	0	0	0

PM10	2018	108	63	2	0	0
PM2.5	2018	105	61	3	0	0
C6H6	2018	88	0	0	0	0
H2S	2019	334	0	0	0	0
SO2	2019	365	0	0	0	0
NOX	2019	314	48	0	0	0
PM10	2019	104	70	5	1	1
PM2.5	2019	114	56	3	3	0
C6H6	2019	60	0	0	0	0
NOX	2020	328	36	1	0	0
SO2	2020	366	0	0	0	0
H2S	2020	351	0	0	0	0
PM10	2020	148	47	1	2	0
PM2.5	2020	138	38	5	1	0
C6H6	2020	53	1	0	0	0
NOX	2021	334	29	0	0	0
H2S	2021	353	0	0	0	0
SO2	2021	364	0	0	0	0
C6H6	2021	79	1	0	0	0
PM2.5	2021	239	43	1	2	0
PM10	2021	149	43	2	0	0
NOX	2022	316	45	1	0	0
H2S	2022	364	0	0	0	0
SO2	2022	365	0	0	0	0
PM10	2022	120	60	4	1	0
C6H6	2022	102	0	0	0	0
PM2.5	2022	249	53	6	2	0
NOX	2023	339	22	0	0	0

	SO2	2023	364	0	0	0	0
	H2S	2023	365	0	0	0	0
	PM2.5	2023	265	48	5	2	0
	C6H6	2023	105	4	0	0	0
	PM10	2023	152	35	1	0	0
Vandellòs i l'Hospitalet de l'Infant (viver)	SO2	2013	352	0	0	0	0
	PM10	2013	253	7	2	0	0
	SO2	2014	364	0	0	0	0
	PM10	2014	212	58	4	1	0
	SO2	2015	359	0	0	0	0
	NOX	2015	201	5	0	0	0
	PM10	2015	241	59	7	3	0
	NOX	2016	342	6	0	0	0
	SO2	2016	365	0	0	0	0
	PM10	2016	289	30	1	2	0
	SO2	2017	362	0	0	0	0
	NOX	2017	359	1	0	0	0
	PM10	2017	273	51	1	2	0
	SO2	2018	365	0	0	0	0
	NOX	2018	322	1	0	0	0
	PM10	2018	273	50	3	0	0
	NOX	2019	364	0	0	0	0
	SO2	2019	364	0	0	0	0
	PM10	2019	240	70	0	9	0
	NOX	2020	358	1	0	0	0
	SO2	2020	360	0	0	0	0
	PM10	2020	257	45	2	2	0
	SO2	2021	357	0	0	0	0

	NOX	2021	332	4	0	0	0
	PM10	2021	268	55	2	1	0
	SO2	2022	365	0	0	0	0
	NOX	2022	360	4	0	0	0
	PM10	2022	259	71	1	1	0
	NOX	2023	362	3	0	0	0
	SO2	2023	362	0	0	0	0
	PM10	2023	283	46	2	3	0
Vila-seca (IES Vila-seca)	H2S	2020	99	0	0	0	0
	PM2.5	2020	66	25	6	2	0
	SO2	2020	99	0	0	0	0
	NOX	2020	81	15	0	0	0
	PM10	2020	82	17	0	0	0
	O3	2020	51	48	0	0	0
	C6H6	2020	12	0	0	0	0
	PM10	2020	32	17	0	3	0
	SO2	2021	365	0	0	0	0
	NOX	2021	339	25	1	0	0
	O3	2021	77	286	2	0	0
	PM10	2021	256	91	9	4	0
	H2S	2021	365	0	0	0	0
	PM2.5	2021	200	122	17	20	1
	C6H6	2021	51	0	0	0	0
	PM10	2021	141	50	9	1	0
	O3	2022	112	234	0	0	0
	PM10	2022	109	24	8	1	0
	PM2.5	2022	79	41	6	9	0
	SO2	2022	362	0	0	0	0

NOX	2022	312	41	4	0	0
H2S	2022	363	0	0	0	0
PM10	2022	99	83	8	1	0
C6H6	2022	54	0	0	0	0
PM10	2023	232	69	2	1	0
PM2.5	2023	229	69	3	3	0
H2S	2023	355	0	0	0	0
NOX	2023	333	19	3	0	0
SO2	2023	333	0	0	0	0
O3	2023	91	267	0	0	0
C6H6	2023	54	0	0	0	0
PM10	2023	129	60	4	0	0

Appendix D. Correlation between air pollution and pediatrics admissions

a) Daily without log transformation

Pollutant	Variable	All		J100		Pande		J100_Pande	
		S	P	S	P	S	P	S	P
PM10	score	-0,072	-0,075	-0,0562	-0,06216	-0,081	-0,037357	-0,065	-0,03128
	pvalue	1,54E-05	8,14E-06	0,002	0,0006	0,0001	0,07671	0,004	0,16975
NOX	score	0,147	0,0438	0,0834	-0,0028	0,21	0,2141	0,116	0,1232
	pvalue	3,89E-19	1,01E-02	5,71E-06	8,79E-01	2,62E-23	1,67E-24	3,05E-07	5,87E-08
PM2.5	score	-0,135	-0,0099	-0,075	0,0243	0,0565	0,00846	-0,025	-0,03214
	pvalue	7,31E-05	7,72E-01	0,108	0,6017	0,38	0,8955	0,778	0,71446
SO2	score	-0,0285	0,0136	0,003	0,005	0,0255	-0,006	-0,06	-0,07357
	pvalue	0,2838	0,609	0,93	0,88442	0,537	0,8831	0,27	0,1885

b) Weekly without log transformation

Pollutant	Variable	All		J100		Pande		J100_Pande	
		S	P	S	P	S	P	S	P
PM10	score	0,0108	-0,0696	0,0333	-0,0498	-0,041	0,01499	0,012	0,0609
	pvalue	7,94E-01	9,56E-02	0,428	2,37E-01	0,43	7,74E-01	0,82	0,2422
NOX	score	0,379	0,1346	0,408	0,106	0,495	0,4238	0,51	0,4641
	pvalue	6,83E-21	1,00E-03	4,92E-24	1,20E-02	4,94E-24	2,15E-17	8,80E-26	6,74E-21
PM2.5	score	-0,093	-0,0357	-0,0859	-0,00889	0,0328	0,024764	-0,13	-0,088
	pvalue	8,49E-02	5,09E-01	0,1694	0,8871	0,667	0,74637	0,18	0,3475
SO2	score	0,0123	0,0219	0,00853	0,01855	0,102	-0,03422	0,08	-0,02
	pvalue	0,7918	0,6389	0,8679	0,7177	0,0889	0,5705	0,25	0,76965

c) Daily with log transformation

Pollutant	Variable	All		J100		Pande		J100_Pande	
		S	P	S	P	S	P	S	P
PM10	score	-0,0779	-0,10288	-0,06	-0,0767	-0,0817	-0,107	-0,0666	-0,0732
	pvalue	3,68E-06	9,74E-10	0,0009	2,62E-05	0,000105	3,69E-07	3,42E-03	0,00129
NOX	score	0,1402	0,03787	0,0759	-0,00749	0,20342	0,1798	0,109	0,096278
	pvalue	1,35E-16	2,60E-02	3,64E-05	6,84E-01	3,13E-22	1,19E-17	1,63E-02	2,32E-05
PM2.5	score	-0,13813	-0,002265	-0,07759	0,02423	0,0565	0,03332	-0,0248	-0,034159
	pvalue	5,01E-05	9,47E-01	0,095	0,603	0,3802	0,6052	7,78E-01	0,69738
SO2	score	-0,05895	0,0018	-0,0247	-0,00147	0,0002513	-0,01454	-0,0801	-0,079
	pvalue	0,0269	0,9453	0,4747	0,966	0,9951	0,7258	0,1519	0,1575

d) Weekly with log transformation

Pollutant	Variable	All		J100		Pande		J100_Pande	
		S	P	S	P	S	P	S	P
PM10	score	0,01088	-0,08257	0,0334	-0,0584	-0,0408	-0,0415	0,0116	0,003
	pvalue	7,95E-01	4,80E-02	0,428	0,16539	0,433	0,4254	0,82	0,952
NOX	score	0,37945	0,1643	0,4089	0,1355	0,49512	0,4519	0,512	0,4897
	pvalue	6,83E-21	8,32E-05	4,92E-24	1,29E-03	4,94E-24	8,06E-20	8,80E-26	2,05E-23
PM2.5	score	-0,093	-0,018	-0,0859	0,01068	0,0328	0,0589	-0,1264	-0,088477
	pvalue	8,50E-02	7,31E-01	0,169477	0,864	0,667726	0,44	0,1779	0,347
SO2	score	0,012353	0,0551	0,0085	0,043	0,1024	0,02274	0,07926	0,0246
	pvalue	0,7918	0,2386	0,8684	0,4	0,0889	0,706	0,246	0,7187

Appendix E. Python code for the air pollution analysis

Xarxa de Vigilància i Previsió de la Contaminació Atmosfèrica - Initial data exploration v3

Dídac Roda Pitarg

Install PySpark

```
import os # install apache spark
os.system("wget -q https://www-us.apache.org/dist/spark/spark-2.4.5/sp
ark-2.4.5-bin-hadoop2.7.tgz")
os.system("tar xf /spark-2.4.5-bin-hadoop2.7.tgz")

!pip install -q pyspark
!apt-get install openjdk-8-jdk-headless -qq



# export environment java path
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"

# Import Pyspark and create session
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

Reach Dades Obertes API

Automatic Data

```
import requests

# fetch automatic data

req = requests.get('https://analisi.transparenciacatalunya.cat/resourc
e/tasf-thgu.csv?$query=SELECT%0A%20%20%60codi_eoi%60%2C%0A%20%20%60nom
_estacio%60%2C%0A%20%20%60data%60%2C%0A%20%20%60magnitud%60%2C%0A%20%2
0%60contaminant%60%2C%0A%20%20%60unitats%60%2C%0A%20%20%60tipus_estaci
o%60%2C%0A%20%20%60area_urbana%60%2C%0A%20%20%60codi_ine%60%2C%0A%20%2
0%60municipi%60%2C%0A%20%20%60codi_comarca%60%2C%0A%20%20%60nom_comarc
a%60%2C%0A%20%20%60h01%60%2C%0A%20%20%60h02%60%2C%0A%20%20%60h03%60%2C
%0A%20%20%60h04%60%2C%0A%20%20%60h05%60%2C%0A%20%20%60h06%60%2C%0A%20%
20%60h07%60%2C%0A%20%20%60h08%60%2C%0A%20%20%60h09%60%2C%0A%20%20%60h1
0%60%2C%0A%20%20%60h11%60%2C%0A%20%20%60h12%60%2C%0A%20%20%60h13%60%2C
%0A%20%20%60h14%60%2C%0A%20%20%60h15%60%2C%0A%20%20%60h16%60%2C%0A%20%
20%60h17%60%2C%0A%20%20%60h18%60%2C%0A%20%20%60h19%60%2C%0A%20%20%60h2
0%60%2C%0A%20%20%60h21%60%2C%0A%20%20%60h22%60%2C%0A%20%20%60h23%60%2C
%0A%20%20%60h24%60%2C%0A%20%20%60altitud%60%2C%0A%20%20%60latitud%60%2
C%0A%20%20%60longitud%60%2C%0A%20%20%60geocoded_column%60%2C%0A%20%20%
60%3A%40computed_region_bh64_c7uy%60%2C%0A%20%20%60%3A%40computed_regi
on_wvic_k925%60%0AWHERE%0A%20%20caseless_one_of(%0A%20%20%20%20%60nom_
estacio%60%2C%0A%20%20%20%20%22Alcover%22%2C%0A%20%20%20%20%22Constant
%C3%AD%22%2C%0A%20%20%20%20%22Perafort%20(Puigdelf%C3%AD)%22%2C%0A%20%
20%20%20%22Reus%22%2C%0A%20%20%20%20%22Tarragona%20(Bonavista)%22%2C%0
A%20%20%20%20%22Tarragona%20(Parc%20de%20la%20Ciutat)%22%2C%0A%20%20%2
0%20%22Tarragona%20(Sant%20Salvador)%22%2C%0A%20%20%20%20%22Tarragona%
20(Universitat%20Laboral)%22%2C%0A%20%20%20%20%22Vila-seca%20(IES%20Vi
```

```
la-seca)%22%2C%0A%20%20%20%22Vandell%C3%B2s%20(Viver)%22%0A%20%20)%
0A%20%20AND%20(%60data%60%0A%20%20%20%20%20%20%20%20BETWEEN%20%2220
13-01-01T00%3A00%3A00%22%20%3A%3A%20floating_timestamp%0A%20%20%20%20
%20%20%20%20AND%20%222023-12-31T23%3A45%3A00%22%20%3A%3A%20floating_
timestamp)%0AORDER%20BY%20%60data%60%20DESC%20NULL%20FIRST%20LIMIT%205
00000')

# Write API response in a CSV file

file_content = req.content

csv_file = open('tasf-thgu.csv', 'wb')
csv_file.write(file_content)
csv_file.close()

# Import CSV as Spark dataframe

archivo = 'tasf-thgu.csv'
df_spark = spark.read.csv(archivo, inferSchema=True, header=True)

df_spark.count() # rows count

df_spark.printSchema() # dataframe columns exploration

df_spark.offset(200).limit(4).toPandas().head() # see df structure

df_spark.orderBy('DATA').limit(4).toPandas().head() # check first reco
rds date

# filter by rows from 2013

from pyspark.sql.functions import year

df_spark = df_spark.filter((year("data") >= 2013) & ((year("data") <=
2023)))

df_spark.count() # rows count
```

Manual Data

```
# fetch manual data

req2 = requests.get("https://analisi.transparenciacatalunya.cat/resour
ce/qg74-87s9.csv?$query=SELECT%0A%20%20%60codi_eoi%60%2C%0A%20%20%60no
m_estacio%60%2C%0A%20%20%60ano%60%2C%0A%20%20%60mes%60%2C%0A%20%20%60m
agnitud%60%2C%0A%20%20%60nom_contaminant%60%2C%0A%20%20%60unitats%60%2
C%0A%20%20%60tipus_estacio%60%2C%0A%20%20%60codi_ine%60%2C%0A%20%20%60
nom_municipi%60%2C%0A%20%20%60d01%60%2C%0A%20%20%60d02%60%2C%0A%20%20%
60d03%60%2C%0A%20%20%60d04%60%2C%0A%20%20%60d05%60%2C%0A%20%20%60d06%6
0%2C%0A%20%20%60d07%60%2C%0A%20%20%60d08%60%2C%0A%20%20%60d09%60%2C%0A
%20%20%60d10%60%2C%0A%20%20%60d11%60%2C%0A%20%20%60d12%60%2C%0A%20%20%
60d13%60%2C%0A%20%20%60d14%60%2C%0A%20%20%60d15%60%2C%0A%20%20%60d16%6
0%2C%0A%20%20%60d17%60%2C%0A%20%20%60d18%60%2C%0A%20%20%60d19%60%2C%0A
%20%20%60d20%60%2C%0A%20%20%60d21%60%2C%0A%20%20%60d22%60%2C%0A%20%20%
60d23%60%2C%0A%20%20%60d24%60%2C%0A%20%20%60d25%60%2C%0A%20%20%60d26%6
0%2C%0A%20%20%60d27%60%2C%0A%20%20%60d28%60%2C%0A%20%20%60d29%60%2C%0A
%20%20%60d30%60%2C%0A%20%20%60d31%60%2C%0A%20%20%60altitud%60%2C%0A%20
%20%60latitud%60%2C%0A%20%20%60longitud%60%2C%0A%20%20%60georefer_ncia
%60%2C%0A%20%20%60%3A%40computed_region_bh64_c7uy%60%2C%0A%20%20%60%3A
%40computed_region_wvic_k925%60%0AWHERE%0A%20%20caseless_one_of(%0A%20
%20%20%20%60nom_estacio%60%2C%0A%20%20%20%20%22Constant%C3%AD%20(Gaud%
C3%AD)%22%2C%0A%20%20%20%20%22El%20Morell%20(Deixalleria%20municipal)%
```

```
22%2C%0A%20%20%20%20%22La%20Canonja%20(deixalleria%20municipal)%22%2C%
0A%20%20%20%20%22Reus%20(el%20Tallapedra)%22%2C%0A%20%20%20%20%22Tarra
gona%20(parc%20de%20la%20Ciutat)%22%2C%0A%20%20%20%20%22Tarragona%20(S
alut)%22%2C%0A%20%20%20%20%22Tarragona%20(Sant%20Salvador)%22%2C%0A%20
%20%20%20%22Tarragona%20(Universitat%20Laboral)%22%2C%0A%20%20%20%20%2
2Vila-seca%20(IES%20Vila-seca)%22%2C%0A%20%20%20%20%22Vandell%C3%B2s%2
0i%20l'Hospitalet%20de%20l'Infant%20(viver)%22%0A%20%20)%0A%20%20AND%2
0((%60ano%60%20BETWEEN%202013%20AND%202023)%0A%20%20%20%20%20%20%20%20
%20AND%20caseless_one_of(%0A%20%20%20%20%20%20%20%20%20%20%20%20%60nom_co
ntaminant%60%2C%0A%20%20%20%20%20%20%20%20%20%20%20%22PM2.5%22%2C%0A%2
0%20%20%20%20%20%20%20%20%20%20%22PM10%22%2C%0A%20%20%20%20%20%20%20%2
0%20%20%20%22C6H6%22%0A%20%20%20%20%20%20%20%20%20))%20LIMIT%20500000"
)

# Write API response in a CSV file

file2_content = req2.content

csv2_file = open('qg74-87s9.csv', 'wb')
csv2_file.write(file2_content)
csv2_file.close()

# Import CSV as Spark dataframe

archivo2 = 'qg74-87s9.csv'
df2_spark = spark.read.csv(archivo2, inferSchema=True, header=True)

df2_spark.count() # rows count

df2_spark.printSchema() # dataframe columns exploration

df2_spark.offset(200).limit(4).toPandas().head() # see df structure

{"type":"dataframe"}

df2_spark.orderBy('ano', 'mes').limit(4).toPandas().head() # check fir
st records date

{"type":"dataframe"}

# filter by rows from 2013

df2_spark = df2_spark.filter((df2_spark.ano >= 2013) & (df2_spark.ano
<= 2023))

df2_spark.count()
```

Mean values per year, contaminant and station

Automatic Data

```
from pyspark.sql.functions import col
df_spark.select(col("nom_estacio")).distinct().collect()

[Row(nom_estacio='Reus'),
 Row(nom_estacio='Perafort (Puigdelfí)'),
 Row(nom_estacio='Tarragona (Universitat Laboral)'),
 Row(nom_estacio='Constantí'),
 Row(nom_estacio='Tarragona (Sant Salvador)'),
 Row(nom_estacio='Vandellòs (Viver)'),
 Row(nom_estacio='Vila-seca (IES Vila-seca)'),
```

```
 Row(nom_estacio='Tarragona (Parc de la Ciutat)'),
 Row(nom_estacio='Tarragona (Bonavista)'),
 Row(nom_estacio='Alcover')]

# rename Vandellòs (Viver) to Vandellòs i l'Hospitalet de l'Infant (vi
ver) to match manual data
from pyspark.sql.functions import *

df_spark = df_spark.withColumn('nom_estacio',when(df_spark.nom_estacio
 != "Vandellòs (Viver)",df_spark.nom_estacio).otherwise("Vandellòs i l
'Hospitalet de l'Infant (viver)"))

from pyspark.sql.functions import col
import matplotlib.pyplot as plt
import pandas as pd

# Convert 'data' column to year
df_new = df_spark.withColumn("year", col("data").substr(1, 4))

# Group by 'nom_estacio', 'contaminant', and 'year' and calculate mean
 for h01 to h24
grouped_df = df_new.groupBy("nom_estacio", "contaminant", "year").agg(
    {"h01": "mean", "h02": "mean", "h03": "mean", "h04": "mean",
     "h05": "mean", "h06": "mean", "h07": "mean", "h08": "mean",
     "h09": "mean", "h10": "mean", "h11": "mean", "h12": "mean",
     "h13": "mean", "h14": "mean", "h15": "mean", "h16": "mean",
     "h17": "mean", "h18": "mean", "h19": "mean", "h20": "mean",
     "h21": "mean", "h22": "mean", "h23": "mean", "h24": "mean"}
)



# Create a new DataFrame with the correct column names
mean_cols = [f"avg_h{i:02d}" for i in range(1, 25)]
columns = ["nom_estacio", "contaminant", "year"] + mean_cols
grouped_df = grouped_df.toDF(*columns)

# Calculate the mean of all avg_hXX columns
grouped_df = grouped_df.selectExpr("*", f"({'+'.join(mean_cols)})/{len
(mean_cols)} as mean")

# Select relevant columns for plotting
plot_df = grouped_df[["nom_estacio", "contaminant", "year", "mean"]]
```

Manual Data

```
df2_spark.select(col("nom_estacio")).distinct().collect()

[Row(nom_estacio='Tarragona (parc de la Ciutat)'),
 Row(nom_estacio='Tarragona (Universitat Laboral)'),
 Row(nom_estacio='El Morell (Deixalleria municipal)'),
 Row(nom_estacio='Tarragona (Sant Salvador)'),
 Row(nom_estacio='Vila-seca (IES Vila-seca)'),
 Row(nom_estacio='Reus (el Tallapedra)'),
 Row(nom_estacio='Constantí (Gaudí)'),
 Row(nom_estacio='Tarragona (Salut)'),
 Row(nom_estacio='La Canonja (deixalleria municipal)'),
 Row(nom_estacio="Vandellòs i l'Hospitalet de l'Infant (viver)")]

# Group by 'nom_estacio', 'contaminant', and 'year' and calculate mean
 for d01 to d31
grouped2_df = df2_spark.groupBy("nom_estacio", "nom_contaminant", "ano
```

```
").agg(
    {"d01": "mean", "d02": "mean", "d03": "mean", "d04": "mean",
     "d05": "mean", "d06": "mean", "d07": "mean", "d08": "mean",
     "d09": "mean", "d10": "mean", "d11": "mean", "d12": "mean",
     "d13": "mean", "d14": "mean", "d15": "mean", "d16": "mean",
     "d17": "mean", "d18": "mean", "d19": "mean", "d20": "mean",
     "d21": "mean", "d22": "mean", "d23": "mean", "d24": "mean",
     "d25": "mean", "d26": "mean", "d27": "mean", "d28": "mean",
     "d29": "mean", "d30": "mean", "d31": "mean"}
)

# Create a new DataFrame with the correct column names
mean2_cols = [f"avg_d{i:02d}" for i in range(1, 32)]
columns2 = ["nom_estacio", "nom_contaminant", "ano"] + mean2_cols
grouped2_df = grouped2_df.toDF(*columns2)

# Calculate the mean of all avg_hXX columns
grouped2_df = grouped2_df.selectExpr("*", f"({'+'.join(mean2_cols)})/{
len(mean2_cols)} as mean")

# Select relevant columns for plotting
plot2_df = grouped2_df[["nom_estacio", "nom_contaminant", "ano", "mean
"]]

# Change columns name to match automatic data format
plot2_df = plot2_df.withColumnRenamed("nom_contaminant", "contaminant"
)\
       .withColumnRenamed("ano", "year")

plot2_df.filter(col('contaminant') == 'PM10').orderBy('year').show(100
)



# merge automatic and manual data into a new df

merged_df = plot_df.union(plot2_df)

import matplotlib.cm as cm
import numpy as np

# Convert PySpark DataFrame to Pandas DataFrame for plotting
pandas_plot_df = merged_df.toPandas()

# Plotting
for contaminant in pandas_plot_df["contaminant"].unique():
    plt.figure(figsize=(10, 6))

    # Get a list of unique station names
    unique_stations = pandas_plot_df["nom_estacio"].unique()

    # Generate a list of colors using a colormap
    colors = cm.get_cmap('tab20', len(unique_stations))

    for idx, nom_estacio in enumerate(unique_stations):
        subset_df = pandas_plot_df[(pandas_plot_df["contaminant"] == c
ontaminant) & (pandas_plot_df["nom_estacio"] == nom_estacio)]

        # Convert 'year' column to numeric type
        subset_df["year"] = pd.to_numeric(subset_df["year"])

        # Sort the DataFrame by 'year' in ascending order
```

```
        subset_df = subset_df.sort_values(by="year")

        plt.plot(subset_df["year"], subset_df["mean"], label=f"{nom_es
tacio}", color=colors(idx))

    plt.xlabel("Year")
    plt.ylabel(f"Mean Value ({contaminant})")
    plt.title(f"Mean {contaminant} Levels per Year")
    plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
    #plt.tight_layout()  # Adjust the plot to make room for the legend
    plt.show()
```

Animated Map

```
from pyspark.sql.functions import month, coalesce, lit, sum
from pyspark.sql import functions as F
from functools import reduce

df_month_new = df_new.withColumn('month', month(df_new['data']))

grouped_month_df = df_month_new.groupBy("nom_estacio", "contaminant",
"year", "month").agg(
    {"h01": "mean", "h02": "mean", "h03": "mean", "h04": "mean",
     "h05": "mean", "h06": "mean", "h07": "mean", "h08": "mean",
     "h09": "mean", "h10": "mean", "h11": "mean", "h12": "mean",
     "h13": "mean", "h14": "mean", "h15": "mean", "h16": "mean",
     "h17": "mean", "h18": "mean", "h19": "mean", "h20": "mean",
     "h21": "mean", "h22": "mean", "h23": "mean", "h24": "mean"}
)

# Create a new DataFrame with the correct column names
mean_month_cols = [f"avg_h{i:02d}" for i in range(1, 25)]
columns_month = ["nom_estacio", "contaminant", "year", "month"] + mean
_month_cols
grouped_month_df = grouped_month_df.toDF(*columns_month)

# Select columns h01 to h24
mean_month_cols = [coalesce(col('avg_h{:02d}'.format(i)), lit(0)) for
i in range(1, 25)]
mean_sum = reduce(lambda x, y: x + y, mean_month_cols)
num_valid_values = reduce(lambda x, y: x + y, [(col(f'avg_h{i:02d}').i
sNotNull()).cast('integer') for i in range(1, 25)])
averageFunc = mean_sum / num_valid_values

grouped_month_df = grouped_month_df.withColumn('mean', averageFunc)

# Select relevant columns for plotting
plot_month_df = grouped_month_df[["nom_estacio", "contaminant", "year"
, "month", "mean"]]

# Show the resulting DataFrame
plot_month_df.show()

from pyspark.sql.functions import col, coalesce, lit

# Group by 'nom_estacio', 'contaminant', and 'year'
grouped2_month_df = df2_spark

# Select columns d01 to d31
mean2_month_cols = [coalesce(col('d{:02d}'.format(i)), lit(0)) for i i
n range(1, 32)]
mean_sum = reduce(lambda x, y: x + y, mean2_month_cols)
```

```
num_valid_values = reduce(lambda x, y: x + y, [(col(f'd{i:02d}').isNot
Null()).cast('integer') for i in range(1, 32)])
averageFunc = mean_sum / num_valid_values

grouped2_month_df = grouped2_month_df.withColumn('mean', averageFunc)

# Change columns name to match automatic data format
plot2_month_df = grouped2_month_df.withColumnRenamed("nom_contaminant"
, "contaminant")\
       .withColumnRenamed("ano", "year")\
       .withColumnRenamed("mes", "month")

# Select relevant columns for plotting
plot2_month_df = plot2_month_df[["nom_estacio", "contaminant", "year",
 "month", "mean"]]
plot2_month_df.show(100)

# merge automatic and manual data into a new df

merged_month_df = plot_month_df.union(plot2_month_df)

# get stations coord

locations = df_spark.select("nom_estacio", "latitud", "longitud").drop
Duplicates()
locations.show()

merged_month_df.select(col("contaminant")).distinct().collect()
merged_month_df = merged_month_df.filter((merged_month_df.contaminant
!= 'PM1') & (merged_month_df.contaminant != 'NO') & (merged_month_df.c
ontaminant != 'NO2'))

merged_month_df.show()

# Convert PySpark DataFrame to Pandas DataFrame for plotting
pandas_month_plot_df = merged_month_df.toPandas()

# left join pandas_plot_df and location
map_df = pandas_month_plot_df.merge(locations.toPandas(), on='nom_esta
cio', how='left')
map_df.head()

# drop rows with nan value in mean, which means the row have all null
values from h01 to h24 or d01 to d31
map_df = map_df.dropna()
map_df.head()

# join month and year columns
map_df['time'] = map_df['month'].astype(str) + "/" + map_df['year'].as
type(str)

map_df.head()

level_list = { 'NOX': {'Good': 0, 'Reasonably good': 41, 'Regular': 91
, 'Unfavourable': 121, 'Very unfavourable': 231, 'Extremely unfavourab
le': 340 },
               'PM10': {'Good': 0, 'Reasonably good': 21, 'Regular': 4
1, 'Unfavourable': 51, 'Very unfavourable': 101, 'Extremely unfavourab
le': 150 },
               'PM2.5': {'Good': 0, 'Reasonably good': 11, 'Regular':
21, 'Unfavourable': 26, 'Very unfavourable': 51, 'Extremely unfavourab
le': 75 },
```

```
                'O3': {'Good': 0, 'Reasonably good': 51, 'Regular': 101
, 'Unfavourable': 131, 'Very unfavourable': 241, 'Extremely unfavourab
le': 380 },
                'SO2': {'Good': 0, 'Reasonably good': 101, 'Regular': 2
01, 'Unfavourable': 351, 'Very unfavourable': 501, 'Extremely unfavour
able': 750 },
                'CO': {'Good': 0, 'Reasonably good': 3, 'Regular': 6, '
Unfavourable': 11, 'Very unfavourable': 21, 'Extremely unfavourable':
50 },
                'C6H6': {'Good': 0, 'Reasonably good': 6, 'Regular': 11
, 'Unfavourable': 21, 'Very unfavourable': 51, 'Extremely unfavourable
': 100 },
                'H2S': {'Good': 0, 'Reasonably good': 26, 'Regular': 51
, 'Unfavourable': 101, 'Very unfavourable': 201, 'Extremely unfavourab
le': 500 },
                }

def get_level(row):
    contaminant = row['contaminant']
    mean = row['mean']

    plevel = ''

    for level, value in level_list[contaminant].items():
        if mean > value:
            plevel = level

    return plevel

# Apply the function to create the 'color' column
map_df['level'] = map_df.apply(get_level, axis=1)

map_df.head()

import plotly.express as px

all_figures = []

level_map = {'Good': 1, 'Reasonably good': 2, 'Regular': 3, 'Unfavoura
ble': 4, 'Very unfavourable': 5, 'Extremely unfavourable': 6}

for contaminant in map_df["contaminant"].unique():
  contaminant_df = map_df[(map_df["contaminant"] == contaminant)]
  print(contaminant)
  contaminant_df['levelNumber']=contaminant_df['level']
  contaminant_df.levelNumber.replace(level_map,inplace=True)
  fig = px.scatter_mapbox(contaminant_df, lat=contaminant_df['latitud'
],
                                lon=contaminant_df['longitud'],
                                size=contaminant_df['mean'],
                                hover_data = 'nom_estacio',
                                animation_frame = 'time',
                                color = 'levelNumber',
                                color_continuous_scale = 'thermal',
                                range_color = (contaminant_df.levelNumbe
r.min(),contaminant_df.levelNumber.max()),
                                #color = 'level',
                                #color_discrete_map = level_map,
                                #category_orders = {"level": list(level_
map.keys())}
                            )
```

```
  fig.update_layout(mapbox_style="carto-positron", mapbox_zoom=10)
  fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
  #fig.update_layout(legend_title_text='ICQA Level')
  fig.update_layout(
    coloraxis_colorbar=dict(
        title="ICQA Level",
      ),
  )
  fig.update_coloraxes(
    colorbar_tickvals = list(level_map.values()),
    colorbar_ticktext = list(level_map.keys()),
  )

  fig.show(config={'scrollZoom': False, 'displayModeBar': False})

  all_figures.append(fig)

with open('p_graph.html', 'w') as f:
    for i, contaminant in enumerate(map_df["contaminant"].unique()):
        f.write(contaminant + "\n")
        f.write(all_figures[i].to_html(full_html=False, include_plotly
js='cdn', auto_play=False))



df_days = df_new

mean_day_cols = [coalesce(col('h{:02d}'.format(i)), lit(0)) for i in r
ange(1, 25)]
mean_day_sum = reduce(lambda x, y: x + y, mean_day_cols)
day_num_valid_values = reduce(lambda x, y: x + y, [(col(f'h{i:02d}').i
sNotNull()).cast('integer') for i in range(1, 25)])
averagedayFunc = mean_day_sum / day_num_valid_values

df_days = df_days.withColumn('mean', averagedayFunc)

df_days.filter(df_days.mean.isNull()).count()

# dump rows with no data
df_days = df_days.filter(df_days.mean.isNotNull())
df_days.show()

from pyspark.sql.functions import stddev, explode, array, struct

h_columns = [f"h{i:02d}" for i in range(1, 25)]

melted_auto_df = df_days.select(
    "contaminant", "nom_estacio", "year",
    explode(array([struct(lit(h).alias("hour"), col(h).cast("float").a
lias("value")) for h in h_columns])).alias("hour_value")
).select("contaminant", "nom_estacio", "year", "hour_value.value")

result_auto = melted_auto_df.groupBy("contaminant", "nom_estacio", "ye
ar") \
                  .agg(stddev("value").alias("stdev"), mean("value").a
lias("mean"))

result_auto.show()

df_man_days = df2_spark

mean_month_cols = [coalesce(col('d{:02d}'.format(i)), lit(0)) for i in
```

```
 range(1, 32)]
mean_month_sum = reduce(lambda x, y: x + y, mean_month_cols)
day_num_valid_values = reduce(lambda x, y: x + y, [(col(f'd{i:02d}').i
sNotNull()).cast('integer') for i in range(1, 32)])
averagedayFunc = mean_month_sum / day_num_valid_values

df_man_days = df_man_days.withColumn('mean', averagedayFunc)

df_man_days.filter(df_man_days.mean.isNull()).count()

# dump rows with no data
df_man_days = df_man_days.filter(df_man_days.mean.isNotNull())
df_man_days.show()

from pyspark.sql.functions import stddev, explode, array, struct

d_columns = [f"d{i:02d}" for i in range(1, 32)]

melted_man_df = df_man_days.select(
    "nom_contaminant", "nom_estacio", "mes", "ano",
    explode(array([struct(lit(h).alias("mes"), col(h).cast("float").al
ias("value")) for h in d_columns])).alias("day_value")
).select("nom_contaminant", "nom_estacio", "ano", "day_value.value")

result_man = melted_man_df.groupBy("nom_contaminant", "nom_estacio", "
ano") \
                  .agg(stddev("value").alias("stdev"), mean("value").a
lias("mean"))

result_man.show()

# make sure both df have the same columns for merging them
result_man = result_man.withColumnRenamed("nom_contaminant", "contamin
ant")\
       .withColumnRenamed("ano", "year")

result_merged = result_auto.union(result_man)
result_merged.show()

from pyspark.sql.functions import concat, round

# Format the stdev and mean
result_merged = result_merged.withColumn("formatted",
                                         concat(round(result_merged.me
an, 2), lit(" ± "), round(result_merged.stdev, 2)))

result_merged = result_merged.filter((result_merged.contaminant != 'PM
1') & (result_merged.contaminant != 'NO') & (result_merged.contaminant
 != 'NO2'))

# Collect the data
grouped = result_merged.groupBy("nom_estacio", "contaminant").pivot("y
ear").agg({"formatted": "first"})

grouped.show()

df_days = df_days.filter((df_days.contaminant != 'PM1') & (df_days.con
taminant != 'NO') & (df_days.contaminant != 'NO2'))

from pyspark.sql.functions import col, mean, expr, when, count, udf
from pyspark.sql.types import StringType
```

```
def categorize_value(mean_value, pollutant):
    categories = level_list[pollutant]
    if mean_value < categories['Reasonably good']:
        return 'Good'
    elif mean_value < categories['Regular']:
        return 'Reasonably good'
    elif mean_value < categories['Unfavourable']:
        return 'Regular'
    elif mean_value < categories['Very unfavourable']:
        return 'Unfavourable'
    elif mean_value < categories['Extremely unfavourable']:
        return 'Very unfavourable'

categorize_udf = udf(categorize_value, StringType())

df_days = df_days.withColumn('category', categorize_udf(col('mean'), c
ol('contaminant')))

#df_days = df_days.withColumn('year', year(df_new['year']))

result_days_df = df_days.groupBy('nom_estacio', 'contaminant', 'year')
.pivot('category').agg(count('*')).fillna(0)

result_days_df.show()

categorize_value_udf = udf(lambda value, pollutant: categorize_value(v
alue, pollutant), StringType())

# List of day columns
day_columns = [f'd{str(i).zfill(2)}' for i in range(1, 32)]

# Melting the DataFrame to a long format to simplify processing
df_long = df_man_days.selectExpr(
    "codi_eoi", "nom_estacio", "ano", "mes", "magnitud", "nom_contamin
ant", "unitats", "tipus_estacio",
    "codi_ine", "nom_municipi", "altitud", "latitud", "longitud", "geo
refer_ncia",
    "stack(31, " + ", ".join([f"'{col}', {col}" for col in day_columns
]) + ") as (day, value)"
)

# Filter out NULL values
df_long = df_long.filter(df_long.value.isNotNull())

# Categorize values
df_long = df_long.withColumn(
    'category',
    categorize_value_udf(col('value'), col('nom_contaminant'))
)

# Count the occurrences of each category
df_counts = df_long.groupBy("nom_estacio", "nom_contaminant", "ano", "
category").agg(count("*").alias("count"))
df_counts.show()

# Pivot the results to get counts for each category in separate column
s
df_pivot = df_counts.groupBy("nom_estacio", "nom_contaminant", "ano").
pivot("category").agg(expr("coalesce(first(count))")).fillna(0)
```

```
# Show final result
df_pivot.show()

df_pivot = df_pivot.drop('null')

result_days_df.show()

# make sure both df have the same columns for merging them
df_pivot = df_pivot.withColumnRenamed("nom_contaminant", "contaminant"
)\
        .withColumnRenamed("ano", "year")

result_days_df = result_days_df.union(df_pivot)

result_days_df.toPandas().sort_values(['nom_estacio', 'year'])
```

Build a daily df

```
daily_auto_df = df_days.select(["contaminant", "data", "nom_estacio",
"mean"])
daily_auto_df.show()

from pyspark.sql.functions import to_timestamp
# Selecting the necessary columns
daily_man_df = df2_spark.select("nom_contaminant", "nom_estacio", "ano
", "mes", *[f"d{str(i).zfill(2)}" for i in range(1, 32)])

# Melting the d01 to d31 columns into rows
daily_man_df = daily_man_df.selectExpr("nom_contaminant", "nom_estacio
", "ano", "mes", "stack(31, " +
                                       ", ".join([f"'d{str(i).zfill(2)}',
d{str(i).zfill(2)}" for i in range(1, 32)]) +
                                       ") as (day, mean)")

# Filtering out rows where 'mean' is NULL
daily_man_df = daily_man_df.filter(col("mean").isNotNull())

# Creating the 'data' column
daily_man_df = daily_man_df.withColumn("data", expr("make_date(ano, me
s, substring(day, 2, 2))"))

# Selecting the final columns
daily_man_df = daily_man_df.select("nom_contaminant", "data", "nom_est
acio", "mean")

daily_man_df = daily_man_df.withColumn('data', to_timestamp('data').ca
st('string'))

# Show the final DataFrame
daily_man_df.show()

merged_man_auto_df = daily_auto_df.union(daily_man_df)
merged_man_auto_df.show()

merged_man_auto_df = merged_man_auto_df.withColumn('nom_estacio', spli
t(col('nom_estacio'), ' \\(').getItem(0))
merged_man_auto_df.show()

merged_man_auto_df.count()

merged_man_auto_df = merged_man_auto_df.groupBy("contaminant", "data",
 "nom_estacio").agg(avg("mean").alias("mean"))
merged_man_auto_df.count()
```

```
merged_man_auto_df.toPandas().to_csv("daily_pollution.csv")
```

Export Notebook to HTML file

```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
```

```
from google.colab import drive
drive.mount("/content/drive")
```

```
!jupyter nbconvert --to html "/content/drive/MyDrive/Colab Notebooks/X
VPCA Data v3.ipynb"
```

Appendix F. Python code for the clinical dataset analysis

Initial data exploration of Reus' hospital dataset

Dídac Roda Pitarg

Install Pyspark

```
import os # install apache spark
os.system("wget -q https://www-us.apache.org/dist/spark/spark-2.4.5/sp
ark-2.4.5-bin-hadoop2.7.tgz")
os.system("tar xf /spark-2.4.5-bin-hadoop2.7.tgz")

!pip install -q pyspark
!apt-get install openjdk-8-jdk-headless -qq

# export environment java path
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"

# Import Pyspark and create session
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
spark

import pandas as pd
```

Load Data

```
pd1 = pd.read_excel('URG_PED_Respiratories FINAL.xlsx', sheet_name=0)
pd2 = pd.read_excel('URG_PED_Respiratories FINAL.xlsx', sheet_name=1)
pd3 = pd.read_excel('URG_PED_Respiratories FINAL.xlsx', sheet_name=3)
pd4 = pd.read_excel('URG_PED_Respiratories FINAL.xlsx', sheet_name=4)
df1 = spark.createDataFrame(pd1)
df2 = spark.createDataFrame(pd2)
df3 = spark.createDataFrame(pd3)
df4 = spark.createDataFrame(pd4)
```

Handle missing values

```
import missingno as msno

msno.matrix(pd1)

msno.matrix(pd2)

msno.matrix(pd3)

msno.matrix(pd4)
```

Litwise deletion (dropping rows)

```
#Drop rows which contains any NaN or missing value for city
pd1.dropna(subset=['postal'],how='any',inplace=True)
print(pd1['postal'].isnull().sum())

pd3.dropna(subset=['descr'],how='any',inplace=True)
print(pd3['descr'].isnull().sum())

pd4.dropna(subset=['descr'],how='any',inplace=True)
print(pd4['descr'].isnull().sum())
```

```
#drop empty column
pd4.drop(columns=['desc_dg'])

# pandas to spark
df1 = spark.createDataFrame(pd1)
df2 = spark.createDataFrame(pd2)
df3 = spark.createDataFrame(pd3)
df4 = spark.createDataFrame(pd4)

from pyspark.sql.functions import round
# prepare to merge the 4 df

# first convert edat from double to int in df1 and df3
df1 = df1.withColumn("Edat", round(df1["Edat"]).cast('integer'))
df3 = df3.withColumn("Edat", round(df3["Edat"]).cast('integer'))

#treat df1
df1 = df1.drop(*['Data Naixement', 'NHC'])
df1.show()

df1.groupBy(['DiagnosticPrincipal', 'DP_Descripcio']).count().select([
'DiagnosticPrincipal', 'DP_Descripcio']).collect()

# map diagnose cie-9 to cie-10
dg_array = {'4660': 'J209', '4808': 'J1289', '48230': 'J154', '460': '
J00', '485': 'J180', '51911': 'J9801',
            '4800': 'J120', '48289': 'J158', '4829': 'J159', '486': 'J
189', '4809': 'J129', '4779': 'J309',
            '48231': 'J154', '4838': 'J168', '48283': 'J156', '4830':
'J157'}

df3.groupBy(['dg', 'desc_dg']).count().select(['dg', 'desc_dg']).colle
ct()

dg_array['5119'] = 'J918'
dg_array['5198'] = 'J22'
dg_array['48239'] = 'J154'
dg_array['4778'] = 'J3089'
dg_array['4772'] = 'J3081'
dg_array['46611'] = 'J210'
dg_array['4802'] = 'J122'
dg_array['4720'] = 'J310'
dg_array['481'] = 'J13'
dg_array['490'] = 'J40'
dg_array['4770'] = 'J301'
dg_array['46619'] = 'J218'

from pyspark.sql.functions import udf, col
from pyspark.sql.types import StringType

def transform_dg(old_dg):
    if dg_array[old_dg]:
      return dg_array[old_dg]

dgtrans_udf = udf(transform_dg, StringType())

df1_new = df1.withColumn('cie10_dg', dgtrans_udf(col('DiagnosticPrinci
pal').cast("string")))

cp_dataset = spark.read.csv('codigos_postales_municipios_entidades.csv
', header=True)
```

```
df1_new = df1_new.join(cp_dataset, df1_new.postal == cp_dataset.codigo
_postal)

df1_new = df1_new.drop(*['postal', 'codigo_postal', 'municipio_id', 'c
odigo_unidad_poblacional', 'entidad_singular_nombre', 'nucleo_nombre']
)

df3_new = df3.withColumn('cie10_dg', dgtrans_udf(col('dg').cast("strin
g")))

# treat df2

# dg = J21.9-Bronquiolitis Aguda No Especificada
# step 1: split in '-'
# step 2: get first item
# step 3: remove '.'

def break_long_dg(dg):
    return dg.split('-')[0].replace('.', '')

dglongbreak_udf = udf(break_long_dg, StringType())

df2_new = df2.withColumn('cie10_dg', dglongbreak_udf(col('Diagnòstic p
rincipal').cast("string")))

# let's get cp from abs pacient

# check first unique ABS pacient
df2_new.groupBy(['ABS pacient']).count().select(['ABS pacient']).colle
ct()

remove_abs = ['0385-Barcelona 8-J', "0178-L'Ametlla De Mar - El Perell
o", "0140-Lleida 6", "0017-Barcelona 1-B",
              "0288-L'Hospitalet De Llobregat 1 - Centre", "0309-Rubi
1", "-", "0328-L'Aldea - Camarles - L'Ampolla",
              "0395-Barcelona 9-H", "0205-Sant Boi De Llobregat 4", "0
266-Viladecans 2", "0000-0000",
              "0120-Flix", "0013-El Vendrell", "0106-Cervera", "0039-B
arcelona 4-A", "0004-Amposta", "0029-Barcelona 2-I",
              "0387-Ripollet 2", "0165-Mora La Nova - Mora D'Ebre", "0
406-Penedes Rural Oest", "0229-Santa Margarida De Montbui",
              "0197-Sabadell 6", "0383-Barcelona  3-H", "0206-Sant Car
les De La Rapita", "0044-Barcelona 5-C", "0113-Deltebre",
              "0082-Berga", "0192-Sabadell 2", "0112-Cornudella De Mon
tsant", "0350-Vic-1 Nord"]

# delete abs from outside camp de tarragona
df2_new = df2_new.filter(col('ABS pacient').isin(remove_abs) == False)
df2_new.count()

df2_new.groupBy(['ABS pacient']).count().select(['ABS pacient']).colle
ct()

def get_cp(abs):
    if "Tarragona" in abs:
        return "Tarragona"

    elif "Reus" in abs:
        return "Reus"

    elif "El Morell" in abs:
      return "El Morell"
```

```
    elif "Vandellos I L'Hospitalet De L'Infant" in abs:
      return "Vandellos I L'Hospitalet De L'Infant"

    elif "Constanti" in abs:
      return "Constanti"

    elif "Riudoms" in abs:
      return "Riudoms"

    elif "Les Borges Del Camp" in abs:
      return "Les Borges Del Camp"

    elif "Mont-Roig Del Camp" in abs:
      return "Mont-Roig Del Camp"

    elif "Salou" in abs:
      return "Salou"

    elif "Falset" in abs:
      return "Falset"

    elif "Montblanc" in abs:
      return "Montblanc"

    elif "Cambrils" in abs:
      return "Cambrils"

    elif "Vila-Seca" in abs:
      return "Vila-Seca"

    elif "Alt Camp Oest" in abs:
      return "Alcover"

    elif "Alt Camp Est" in abs:
      return "Vila-rodona"

    elif "La Selva Del Camp" in abs:
      return "La Selva Del Camp"

    elif "Valls Urba" in abs:
      return "Valls"

    elif "Torredembarra" in abs:
      return "Torredembarra"

cpfromabs_udf = udf(get_cp, StringType())

df2_new = df2_new.withColumn('municipio_nombre', cpfromabs_udf(col('AB
S pacient')))


# select needed columns for each df to merge
newcolumns = ["data_fi", "edad", "sexe", "municipio_nombre", "dg"]

df1_new = df1_new.select(["Data Alta", "Edat", "Sexe", "municipio_nomb
re", "cie10_dg"])
df1_new = df1_new.toDF(*newcolumns)
df1_new.show()

df2_new = df2_new.select(["Data d'alta", "Edat", "Sexe", "municipio_no
mbre", "cie10_dg"])
```

```
df2_new = df2_new.toDF(*newcolumns)
df2_new.show()

df3_new = df3_new.select(["data_fi", "Edat", "sexe", "descr", "cie10_d
g"])
df3_new = df3_new.toDF(*newcolumns)
df3_new.show()

df4 = df4.select(["data_fi", "edad", "sexe", "descr", "dg"])
df4 = df4.toDF(*newcolumns)
df4 = df4.withColumn("sexe", df4["sexe"].cast("string"))
df4 = df4.replace(['0', '1'], ['Home', 'Dona'], 'sexe')
df4.show()

merged_df = df1_new.union(df2_new)
merged_df = merged_df.union(df3_new)
merged_df = merged_df.union(df4)
merged_df.show()

from pyspark.sql.functions import rtrim

# clean whitespaces of dg column
merged_df = merged_df.withColumn("municipio_nombre", rtrim(col("munici
pio_nombre")))
merged_df.select('municipio_nombre').distinct().collect()

import requests
comarques = requests.get('https://api.idescat.cat/emex/v1/nodes.json')

import json
# Convert JSON String to Python
comarques_dict = json.loads(comarques.text)

# Print Dictionary
print(comarques_dict)

altcamp = comarques_dict['fitxes']['v']['v'][0]['v']

baixcamp = comarques_dict['fitxes']['v']['v'][8]['v']

baixpenedes = comarques_dict['fitxes']['v']['v'][12]['v']

concabarbera = comarques_dict['fitxes']['v']['v'][16]['v']

priorat = comarques_dict['fitxes']['v']['v'][31]['v']

tarragones = comarques_dict['fitxes']['v']['v'][38]['v']

from pyspark.sql.functions import array_contains, lit, col, udf
from pyspark.sql.types import BooleanType

def remove_not_camptarragona(municipi):
  for item in altcamp:
    if 'content' in item:
      if municipi.upper() in item['content'].upper():
        return True

  for item in baixcamp:
    if municipi.upper() in item['content'].upper():
      return True

  for item in baixpenedes:
    if municipi.upper() in item['content'].upper():
```

```
      return True

  for item in concabarbera:
    if municipi.upper() in item['content'].upper():
      return True

  for item in priorat:
    if municipi.upper() in item['content'].upper():
      return True

  for item in tarragones:
    if municipi.upper() in item['content'].upper():
      return True

  return False

is_substring_of_any_udf = udf(lambda text: remove_not_camptarragona(te
xt), BooleanType())

merged_df_new = merged_df.filter(is_substring_of_any_udf(col("municipi
o_nombre")))

merged_df_new.count()

merged_df_new.select(col("municipio_nombre")).distinct().collect()

def rename_camptarragona(municipi):
  for item in altcamp:
    if 'content' in item:
      if municipi.upper() in item['content'].upper():
        return str(item['content'])

  for item in baixcamp:
    if municipi.upper() in item['content'].upper():
      return str(item['content'])

  for item in baixpenedes:
    if municipi.upper() in item['content'].upper():
      return str(item['content'])

  for item in concabarbera:
    if municipi.upper() in item['content'].upper():
      return str(item['content'])

  for item in priorat:
    if municipi.upper() in item['content'].upper():
      return str(item['content'])

  for item in tarragones:
    if municipi.upper() in item['content'].upper():
      return str(item['content'])

rename_city_udf = udf(lambda text: rename_camptarragona(text), StringT
ype())

merged_df_new = merged_df_new.withColumn("municipio_nombre", rename_ci
ty_udf(col("municipio_nombre")))

merged_df_new.count()

from pyspark.sql.functions import rtrim
```

```
# clean whitespaces of dg column
merged_df_new = merged_df_new.withColumn("dg", rtrim(col("dg")))

merged_pd = merged_df_new.toPandas()

msno.matrix(merged_pd)

merged_pd.to_csv("urg_cleaned_data.csv", index=False)
```

Descriptive Analysis

```
clean_data = spark.read.csv('urg_cleaned_data.csv', header=True)

clean_data.select("dg").distinct().collect()



import matplotlib.pyplot as plt

sexe_count = clean_data.groupBy(['sexe']).count().toPandas()
sexe_count.head()

plt.pie(sexe_count['count'], labels=sexe_count['sexe'], autopct='%1.1f
%%', pctdistance=1.1, labeldistance=.6)
plt.title('Gender Distribution')

from pyspark.sql.types import IntegerType
clean_data = clean_data.withColumn("edad", clean_data["edad"].cast(Int
egerType()))
age_count = clean_data.groupBy(['edad']).count().orderBy("edad", ascen
ding=False).toPandas()
age_count.head(100)

x = [i for i in range(len(age_count.index))]
plt.bar(age_count['edad'], age_count['count'])
plt.title('Age distribution')
plt.xlabel('Age')
plt.ylabel('Cases')

dg_count = clean_data.groupBy(['dg']).count().orderBy("count", ascendi
ng=False).toPandas()
dg_count.head()

# limit 3 most popular dgs
dg_count = dg_count.head(3)

x = [i for i in range(len(dg_count.index))]
plt.bar(dg_count['dg'], dg_count['count'])
plt.title('3 most common diagnoses')
plt.xlabel('CIE10 code')
plt.ylabel('Cases')

from pyspark.sql.functions import split
muni_dg_year = clean_data
split_col = split(muni_dg_year['data_fi'], '-')
muni_dg_year = muni_dg_year.withColumn("year", split_col.getItem(0))
muni_dg_year.show()
cp_dg_count = muni_dg_year.groupBy(['municipio_nombre', 'dg', 'year'])
.count().orderBy("count", ascending=False).toPandas()
cp_dg_count.head(100)

cp_dg_count_3dg = cp_dg_count.loc[(cp_dg_count['dg'] == 'J00') | (cp_d
g_count['dg'] == 'J209') | (cp_dg_count['dg'] == 'J218')]
```

```
# Plotting
for city in cp_dg_count_3dg["municipio_nombre"].unique():
    plt.figure(figsize=(10, 6))
    for dg in cp_dg_count_3dg["dg"].unique():
        subset_df = cp_dg_count_3dg[(cp_dg_count_3dg["municipio_nombre
"] == city) & (cp_dg_count_3dg["dg"] == dg)]

        # Convert 'year' column to numeric type
        subset_df["year"] = pd.to_numeric(subset_df["year"])

        # Sort the DataFrame by 'year' in ascending order
        subset_df = subset_df.sort_values(by="year")

        plt.plot(subset_df["year"], subset_df["count"], marker='o', la
bel=f"{dg}")

    plt.xlabel("Year")
    plt.ylabel(f"Cases")
    plt.title(f"Cases per Year in {city}")
    plt.legend()
    plt.show()
```

Map cities to stations

```
daily_pollution = spark.read.option("delimiter", ",").option("header",
 True).csv("daily_pollution.csv")

daily_pollution = daily_pollution.drop("_c0")
daily_pollution.show()

daily_pollution.select("nom_estacio").distinct().collect()

daily_clean_data = clean_data
daily_clean_data.select("municipio_nombre").distinct().collect()

from pyspark.sql.functions import to_timestamp
daily_clean_data = daily_clean_data.withColumn('data_fi', to_timestamp
('data_fi').cast('string'))
daily_clean_data.show()

import geopy

from geopy.distance import geodesic as GD
from geopy.geocoders import Nominatim

from geopy.geocoders import Photon
geolocator = Photon(user_agent="measurements", timeout=10)

assigned_stations = {}

all_stations = daily_pollution.select("nom_estacio").distinct().collec
t()
all_cities = daily_clean_data.select("municipio_nombre").distinct().co
llect()

def get_closest_station():
    for city in all_cities:
      location_city2 = geolocator.geocode(city)

      closest_distance = float('inf')
      closest_station = None
```

```
      #if not location_city2:
          #return

      lat_long_city2 = (location_city2.latitude, location_city2.longit
ude)

      for estacio in all_stations:
        location_city1 = geolocator.geocode(estacio)
        #if not location_city1:
            #continue

        lat_long_city1 = (location_city1.latitude, location_city1.long
itude)
        distance = GD(lat_long_city1, lat_long_city2).km

        if distance <= 10 and distance < closest_distance:
            closest_distance = distance
            closest_station = estacio
            assigned_stations[city] = estacio

    return

all_cities =
daily_clean_data.select("municipio_nombre").distinct().collect()

assigned_stations = {}

all_stations =
daily_pollution.select("nom_estacio").distinct().collect()

for city in all_cities:
  location_city2 = geolocator.geocode(city)
  #if not location_city2:
      #return

  closest_distance = float('inf')
  closest_station = None

  lat_long_city2 = (location_city2.latitude, location_city2.longitude)

  for estacio in all_stations:
    location_city1 = geolocator.geocode(estacio)
    #if not location_city1:
        #continue

    lat_long_city1 = (location_city1.latitude,
location_city1.longitude)
    distance = GD(lat_long_city1, lat_long_city2).km

    if distance <= 10 and distance < closest_distance:
        closest_distance = distance
        closest_station = estacio
        assigned_stations[city] = estacio
        print(city, estacio)

get_closest_station()

# Convert dictionary to list of dictionaries
data_list = [{"municipio_nombre": key.municipio_nombre, "nom_estacio":
 value.nom_estacio} for key, value in assigned_stations.items()]
```

```
data_list[11] = {'municipio_nombre': 'Selva del Camp, la',
'nom_estacio': 'Reus'}
data_list[14] = {'municipio_nombre': 'Pobla de Mafumet, la',
'nom_estacio': 'Perafort'}
data_list[20] = {'municipio_nombre': 'Vilallonga del Camp',
'nom_estacio': 'Perafort'}
data_list[25] = {'municipio_nombre': 'Rourell, el', 'nom_estacio':
'Perafort'}


# Create DataFrame
assigned_stations_df = spark.createDataFrame(data_list)

assigned_stations_df.show()

# work only with PM2.5, PM10, SO2, NOx
daily_pollution_filtered = daily_pollution.filter((daily_pollution.con
taminant == "PM2.5") | (daily_pollution.contaminant == "PM10")
                  | (daily_pollution.contaminant == "SO2") | (daily_po
llution.contaminant == "NOX"))
daily_pollution_filtered.show()



from pyspark.sql.functions import col

# Join daily_clean_data and assigned_stations_list
daily_clean_data_renamed = daily_clean_data.withColumnRenamed("municip
io_nombre", "municipio_nombre_2")
daily_stations = daily_clean_data_renamed.join(assigned_stations_df, (
daily_clean_data_renamed.municipio_nombre_2 == assigned_stations_df.mu
nicipio_nombre), how='left')
daily_stations.show()
daily_stations_renamed = daily_stations.withColumnRenamed("nom_estacio
", "nom_estacio_2")

# Join daily_stations and daily_pollution
daily_pollution_stations = daily_stations_renamed.join(daily_pollution
_filtered, (daily_stations_renamed.nom_estacio_2 == daily_pollution_fi
ltered.nom_estacio) & (daily_stations.data_fi == daily_pollution_filte
red.data), how='left')
daily_pollution_stations.show()



daily_pollution_stations.toPandas().to_csv("daily_pollution_stations.c
sv")

daily_pollution_stations = spark.read.csv("daily_pollution_stations.cs
v", header = True)
daily_pollution_stations = daily_pollution_stations.drop(*['_c0'])
daily_pollution_stations.show()



from pyspark.sql.functions import col
daily_pollution_stations_j100 = daily_pollution_stations.filter(col("d
g") == "J00")
daily_pollution_stations_j100.show()
```

```
daily_pollution_stations = daily_pollution_stations.select(["data", "e
dad", "sexe", "municipio_nombre", "nom_estacio", "contaminant", "mean"
])

daily_pollution_stations.show()



daily_pollution_stations_j100 = daily_pollution_stations_j100.select([
"data", "edad", "sexe", "municipio_nombre", "nom_estacio", "contaminan
t", "mean"])

daily_pollution_stations.count()

daily_pollution_stations_j100.count()

daily_pollution_stations = daily_pollution_stations.dropna()
daily_pollution_stations.count()

daily_pollution_stations_j100 = daily_pollution_stations_j100.dropna()
daily_pollution_stations_j100.count()

daily_pollution_stations = daily_pollution_stations.select(["data", "e
dad", "sexe", "municipio_nombre", "nom_estacio", "contaminant", "mean"
])

daily_pollution_stations_j100.show()

daily_pollution_stations_j100 = daily_pollution_stations_j100.select([
"data", "edad", "sexe", "municipio_nombre", "nom_estacio", "contaminan
t", "mean"])

daily_pollution_stations.show()

from scipy.stats import spearmanr, pearsonr
from pyspark.sql.functions import avg, count
import numpy as np

# just leave data, nom_estacio, contaminant and mean
daily_poll_station_urg_corr = daily_pollution_stations.select(["data",
 "nom_estacio", "contaminant", "mean"])
daily_pollution_stations_j100_corr = daily_pollution_stations_j100.sel
ect(["data", "nom_estacio", "contaminant", "mean"])
daily_poll_station_urg_corr_pande = daily_poll_station_urg_corr.filter
("data <= date'2020-01-28'")
daily_poll_station_urg_j100_corr_pande = daily_pollution_stations_j100
_corr.filter("data <= date'2020-01-28'")
for contaminant in daily_poll_station_urg_corr.select(col("contaminant
")).distinct().collect():
  daily_poll_stat_urg = daily_poll_station_urg_j100_corr_pande.filter(
col("contaminant") == contaminant[0])
  daily_poll_stat_urg = daily_poll_stat_urg.select(["data", "mean"])
  print(contaminant[0])
  result_df = daily_poll_stat_urg.groupBy("data").agg(
    avg("mean").alias("mean_avg"),
    count("data").alias("rows_count")
  )
  #result_df.show()
  res = spearmanr(np.array(result_df.select("rows_count").collect()),
np.array(result_df.select("mean_avg").collect()))
  print(res.statistic)
  print('pvalue', res.pvalue)
  pearson = pearsonr(np.array(result_df.select("rows_count").collect()
```

```
).flatten(), np.array(result_df.select("mean_avg").collect()).flatten(
))
  print('pearson coef:',pearson.statistic)
  print('pearson pvalue:', pearson.pvalue)

from pyspark.sql.functions import ln
# daily with log transformation

# just leave data, nom_estacio, contaminant and mean
#daily_poll_station_urg_corr = daily_pollution_stations.select(["data"
, "nom_estacio", "contaminant", "mean"])
#daily_pollution_stations_j100_corr = daily_pollution_stations_j100.se
lect(["data", "nom_estacio", "contaminant", "mean"])
#daily_poll_station_urg_corr_pande = daily_poll_station_urg_corr.filte
r("data <= date'2020-01-28'")
#daily_poll_station_urg_j100_corr_pande = daily_pollution_stations_j10
0_corr.filter("data <= date'2020-01-28'")

for contaminant in daily_poll_station_urg_corr.select(col("contaminant
")).distinct().collect():
  daily_poll_stat_urg = daily_poll_station_urg_j100_corr_pande.filter(
col("contaminant") == contaminant[0])
  daily_poll_stat_urg = daily_poll_stat_urg.select(["data", "mean"])
  daily_poll_stat_urg = daily_poll_stat_urg.withColumn("mean", ln(col(
"mean")))
  print(contaminant[0])
  result_df = daily_poll_stat_urg.groupBy("data").agg(
    avg("mean").alias("mean_avg"),
    count("data").alias("rows_count")
  )
  #result_df.show()
  res = spearmanr(np.array(result_df.select("rows_count").collect()),
np.array(result_df.select("mean_avg").collect()))
  print(res.statistic)
  print('pvalue', res.pvalue)
  pearson = pearsonr(np.array(result_df.select("rows_count").collect()
).flatten(), np.array(result_df.select("mean_avg").collect()).flatten(
))
  print('pearson coef:',pearson.statistic)
  print('pearson pvalue:', pearson.pvalue)

from pyspark.sql.functions import year, weekofyear
from pyspark.sql.functions import sum as _sum

for contaminant in daily_poll_station_urg_corr.select(col("contaminant
")).distinct().collect():
  daily_poll_stat_urg = daily_poll_station_urg_j100_corr_pande.filter(
col("contaminant") == contaminant[0])
  daily_poll_stat_urg = daily_poll_stat_urg.select(["data", "mean"])
  print(contaminant[0])
  result_df = daily_poll_stat_urg.groupBy("data").agg(
    avg("mean").alias("mean_avg"),
    count("data").alias("rows_count")
  )
  #result_df.show(5000)
  result_df = result_df.groupBy(year("data").alias("year"), weekofyear
("data").alias("week")) \
  .agg(avg("mean_avg").alias("mean_avg"), _sum("rows_count").alias("ro
ws_count")) \
  .orderBy("year", "week")
  #result_df.show()
  res = spearmanr(np.array(result_df.select("rows_count").collect()),
```

```
np.array(result_df.select("mean_avg").collect()))
  print(res.statistic)
  print('pvalue', res.pvalue)
  pearson = pearsonr(np.array(result_df.select("rows_count").collect()
).flatten(), np.array(result_df.select("mean_avg").collect()).flatten(
))
  print('pearson coef:',pearson.statistic)
  print('pearson pvalue:', pearson.pvalue)

from pyspark.sql.functions import year, weekofyear, sum as _sum, ln
# weekly with log trans

#daily_poll_station_urg_corr = daily_pollution_stations.select(["data"
, "nom_estacio", "contaminant", "mean"])
#daily_pollution_stations_j100_corr = daily_pollution_stations_j100.se
lect(["data", "nom_estacio", "contaminant", "mean"])
#daily_poll_station_urg_corr_pande = daily_poll_station_urg_corr.filte
r("data <= date'2020-01-28'")
#daily_poll_station_urg_j100_corr_pande = daily_pollution_stations_j10
0_corr.filter("data <= date'2020-01-28'")

for contaminant in daily_poll_station_urg_corr.select(col("contaminant
")).distinct().collect():
  daily_poll_stat_urg = daily_poll_station_urg_j100_corr_pande.filter(
col("contaminant") == contaminant[0])
  daily_poll_stat_urg = daily_poll_stat_urg.select(["data", "mean"])
  print(contaminant[0])
  result_df = daily_poll_stat_urg.groupBy("data").agg(
    avg("mean").alias("mean_avg"),
    count("data").alias("rows_count")
  )
  #result_df.show(5000)
  result_df = result_df.groupBy(year("data").alias("year"), weekofyear
("data").alias("week")) \
  .agg(avg("mean_avg").alias("mean_avg"), _sum("rows_count").alias("ro
ws_count")) \
  .orderBy("year", "week")
  result_df = result_df.withColumn("mean_avg", ln(col("mean_avg")))
  #result_df.show()
  res = spearmanr(np.array(result_df.select("rows_count").collect()),
np.array(result_df.select("mean_avg").collect()))
  print(res.statistic)
  print('pvalue', res.pvalue)
  pearson = pearsonr(np.array(result_df.select("rows_count").collect()
).flatten(), np.array(result_df.select("mean_avg").collect()).flatten(
))
  print('pearson coef:',pearson.statistic)
  print('pearson pvalue:', pearson.pvalue)
```

Machine Learning

```
daily_poll_stat_urg_ml = daily_poll_station_urg_corr_pande.filter(col(
"contaminant") == 'NOX')
daily_poll_stat_urg_ml = daily_poll_stat_urg_ml.select(["data", "mean"
])
result_df_ml = daily_poll_stat_urg_ml.groupBy("data").agg(
  avg("mean").alias("mean_avg"),
  count("data").alias("rows_count")
)
result_df_ml = result_df_ml.groupBy(year("data").alias("year"), weekof
year("data").alias("week")) \
.agg(avg("mean_avg").alias("mean_avg"), _sum("rows_count").alias("rows
```

```
_count")) \
.orderBy("year", "week")
result_df_ml.show()
```

```
from sklearn.svm import SVR
from sklearn.cross_decomposition import PLSRegression

import seaborn as sns
import matplotlib.pyplot as plt

result_df_ml_pd = result_df_ml.drop(*["year", "week"]).toPandas()

corr = result_df_ml_pd.corr(method = 'spearman')

sns.heatmap(corr, annot = True)

plt.show()

from sklearn.model_selection import train_test_split
# Split dataset into training set and test set
"""X_train, X_test, y_train, y_test = train_test_split(result_df_ml_pd
.mean_avg,
                                                       result_df_ml_pd.ro
ws_count,
                                                       test_size=0.2, ran
dom_state=0,
                                                       stratify = result_
df_ml_pd.rows_count)"""
train_df, test_df = result_df_ml.randomSplit(weights=[0.8,0.2], seed=1
00)

X_train = np.array(train_df.select(col("mean_avg")).collect())
y_train = np.array(train_df.select(col("rows_count")).collect())

X_test = np.array(test_df.select(col("mean_avg")).collect())
y_test = np.array(test_df.select(col("rows_count")).collect())

regressor = SVR(kernel = 'rbf')
regressor.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = regressor.predict(X_test)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:11
43: DataConversionWarning: A column-vector y was passed when a 1d arra
y was expected. Please change the shape of y to (n_samples, ), for exa
mple using ravel().
  y = column_or_1d(y, warn=True)
```

```
# Model Accuracy: how often is the classifier correct?
print("Accuracy:",regressor.score(X_test, y_test))

import matplotlib.pyplot as plt

# Plot rows_count vs mean_avg for real data and predicted data
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='blue', label='Real Data')
plt.scatter(X_test, y_pred, color='red', label='Predicted Data')
plt.xlabel('Patients per day')
plt.ylabel('NOX mean per patients and day')
```

```
plt.title('Patients and pollutants per day')
plt.legend()
plt.show()



pls = PLSRegression(n_components=1)
pls.fit(X_train, y_train)

y_pred = pls.predict(X_test)

# Model Accuracy: how often is the classifier correct?
print("Accuracy:",pls.score(X_test, y_test))

Accuracy: 0.2505872384569192

import matplotlib.pyplot as plt

# Plot rows_count vs mean_avg for real data and predicted data
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='blue', label='Real Data')
plt.scatter(X_test, y_pred, color='red', label='Predicted Data')
plt.xlabel('Patients per day')
plt.ylabel('NOX mean per patients and day')
plt.title('Patients and pollutants per day')
plt.legend()
plt.show()

# 5-fold Cross-Validation
# in the training set to calculate validation accuracy

from sklearn.model_selection import cross_val_score
from sklearn import metrics

# SVR
scores = cross_val_score(regressor, X_train, y_train, cv=5)
print(scores)

# PLS

scores = cross_val_score(pls, X_train, y_train, cv=5)
print(scores)
```