# HADO_CARES

***Release 0.1***

**Pablo Villar del Castillo Fernandez**

**Oct 24, 2023**

# CONTENTS

CHAPTER

ONE

# INDEX

## 1.1 Project Documentation

### 1.1.1 1. Introduction

#### 1.1 Context and Problem

In the current era, where technology is advancing at a rapid pace, the realm of healthcare is flooded with a monumental amount of data. While these data have the potential to uncover valuable insights and enhance patient care, their efficient management and analysis pose a significant challenge, especially for health institutions that still largely depend on manual and traditional processes for data management.

Specifically, in the HADO area of the Santiago de Compostela hospital, patient records are managed, among other ways, manually using Excel spreadsheets. This methodology, although functional, leads to a lack of standardization in data formats and limited utilization of the collected information, further hampered by the absence of an efficient system to process and analyze the data in an integrated and cohesive manner.

#### 1.2 Need and Justification for the Project

The core project of this Master's Thesis (TFM) arises in response to this notable gap. It seeks to implement a more sophisticated technological and analytical approach with the goal of maximizing the value obtained from the collected data, providing deeper insights into diagnoses and treatments, and identifying trends that can be crucial for the continuous improvement of patient care.

#### 1.3 Objectives

The main objective of this project is to enhance the current process of patient tracking in HADO by:

- Conducting exploratory data analyses (EDA) with a special focus on main diagnoses.
- Identifying trends and classifying high-cardinality variables.
- Applying Natural Language Processing (NLP) techniques and modeling to group and classify variables.
- Creating an application for the visualization of the transformed and analyzed data, thus assisting in the decision-making process of HADO professionals.

### 1.4 Methodological Approach

The project is developed using the Kedro framework to ensure a reproducible and robust data science workflow and the Streamlit visualization tool, thereby providing an application that serves as an interface for data visualization and results. The complete development process, from data collection and preprocessing to analysis and visualization of results, will be detailed throughout this document.

The code for this project is accessible in the repository: Link: TFM GitHub project.

## 1.1.2 2. Data Problem Description and Analysis

### 2.1 Data Issues

Throughout the EDA and data preprocessing process, various challenges associated with data quality and consistency were identified. Among these, inconsistencies and lack of standardization were particularly prevalent. The records, originating from multiple annual files and being manually managed, presented a series of inconsistencies and shortcomings that demanded detailed cleaning and preprocessing.

The challenge also lay in the data quality. The need to perform extensive cleanings and transformations indicated the existence of problems in the raw data, which could be mitigated in the future through more standardized and automated data collection and management. Furthermore, the presence of null values in various critical variables required a cautious management strategy to prevent the introduction of biases or errors into subsequent analyses.

#### 2.1.1. Implemented Solutions and Strategies

To address these issues, several strategies and solutions were implemented. Data standardization was a crucial step, where a unified and coherent dataset was created through the concatenation and standardization of various annual datasets. Additionally, text processing and data handling techniques were used to clean and transform the variables, thus improving the data quality for future analyses. In particular, a strategy was proposed to handle the NA values, grounded on the nature of each variable and the clinical context.

The use of Kedro facilitated a robust and reproducible data science workflow, ensuring that analyses and models could be easily replicated and audited.

#### 2.1.2. Challenges and Future Considerations

Although measures have been taken to mitigate the identified data problems, there are challenges and considerations for the future. Automating the process, from data collection to the initial preprocessing phases, could improve efficiency and consistency in the future. It is also imperative to develop more sophisticated strategies or predictive models to handle missing data, especially in critical variables. Although this work focused on cleaning and preprocessing, subsequent analysis stages should explore the patterns and relationships in the data more deeply, leading to deeper analyses.

## 2.2 Raw Data Description

### 2.2.1 Data Dictionary

These are the columns for Raw Data:

Table 1: Data Dictionary raw Data

| Column Name | Data Type | Description |
|---|---|---|
| Hospital | Object | Name or identifier of the hospital. |
| Servicio | Object | Service type or department within the hospital. |
| AP | Object | Possibly refers to a specific medical procedure or department. |
| Otros | Object | A category for other miscellaneous entries. |
| Diagnostico | Object | Diagnosis information. |
| Motivo Ing | Object | Reason for admission or inquiry (e.g., symptom control). |
| paliativo Onc | Object | Indicates if palliative care for oncology is provided. |
| Paliativo No Onc | Object | Indicates if palliative care for non-oncology is provided. |
| Fiebre | Object | Indicates presence of fever. |
| Disnea | Object | Indicates presence of shortness of breath. |
| Dolor | Object | Indicates presence of pain. |
| Delirium | Object | Indicates presence of delirium. |
| Astenia | Object | Indicates presence of asthenia (weakness). |
| Anorexia | Object | Indicates presence of anorexia. |
| Otros.1 | Object | Another category for other miscellaneous entries. |
| P terminal | Object | Possibly refers to terminal phase of a condition. |
| Agonía | Object | Indicates presence of agony. |
| PS/IK | Object | Possibly refers to performance status or a specific score/index. |
| Barthel | Object | Possibly refers to the Barthel Index, a measure of disability. |
| GDS-FAST | Object | Possibly refers to the Global Deterioration Scale or Functional Assessment Staging Test. |
| EVA ing | Object | Possibly refers to a type of assessment or score at admission. |
| Otros.2 | Object | Another category for other miscellaneous entries. |
| Complicaciones | Object | Indicates presence of complications. |
| Nº estancias | Float64 | Number of stays or admissions. |
| Nº visitas | Float64/Object | Number of visits. |
| SEDACIÓN | Object | Indicates presence of sedation. |
| Mot. ALTA | Object | Reason for discharge or end of care. |
| Médico | Object | Name or identifier of the medical professional. |
| unnamed.1 | Float64 | Unspecified column with sparse data, likely an error or misplaced data. |

### 2.2.2 Data Features

The datasets vary year by year, not only in terms of the number of entries but also in their structure and quality. Below are the main characteristics of the annual datasets from 2017 to 2022:

- **Inconsistencies in column names:** The datasets exhibit variability in column names, reflecting a lack of standardization in data capture and storage.
- **Variability in Columns:** Some years have additional columns or fewer columns compared to other years, highlighting the need to align and reconcile these differences during preprocessing.
- **Presence of Null Values:** Certain columns, such as "Discharge Date" in 2022 and "Complications" in other years, have a significant number of null values, which requires a considered strategy for handling missing data.

---

**Note:** This code snippet is designed for exploring multiple DataFrames within a Kedro project.

**Requirements**: Ensure that the Kedro project and the related data are executed and available. The names of the DataFrames should be defined in the Kedro catalog. If you are using IDE like VS Code run %load_ext kedro.ipython %load_ext kedro.ipython

---

Listing 1: Exploring Multiple DataFrames

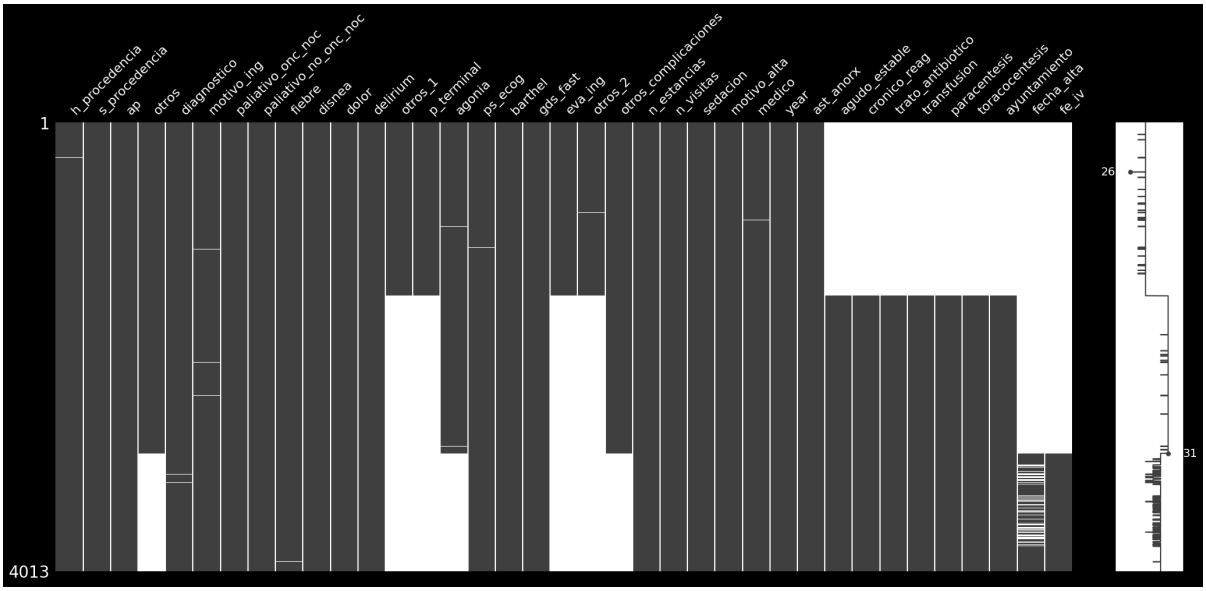
```
# Exploring Multiple DataFrames
# Requirements: The Kedro project and data must be executed, DataFrame names are defined␣
↪in the catalog.

# DataFrame names that you want to explore
dfs_names = ['hado_22', 'hado_21', 'hado_20', 'hado_19', 'hado_18', 'hado_17']

# Loop to print information for each DataFrame
for name in dfs_names:
    # Assuming 'catalog.load' loads the DataFrame based on its name
    # Adjust this line as needed if it's not the case
    df = catalog.load(name)

    print(f"Information and Describe for DataFrame: {name}")
    print("-------------------------------------")

    # Display the DataFrame information
    print(df.info(), df.describe(include='all').T)

    print("\\n\\n")  # Add a couple of blank lines to separate the information from␣
↪different DataFrames
```

### 2.2.3 Data Quality

- **Missing Data:** It is observed that the variable "Discharge Date" is only available for half of 2022, representing a limitation for any temporal analysis involving this variable. It is crucial to investigate the presence of missing data in other variables and manage them adequately to avoid biases in subsequent analyses.

Listing 2: Exploring null values

```
import missingno as msno
df = catalog.load('hado_concat')
msno.matrix(df)
```




- **Data Consistency:** Data consistency will be evaluated by analyzing outlier and unexpected values in the different variables.
- **Variable Cardinality:** Variables like "Main Diagnosis" and "Reason for Admission (ING)" present high cardinality, which complicates analysis. Following strategies like grouping or transforming categories are necessary to handle this complexity.

Listing 3: Pandas profiling Report

```
from ydata_profiling import ProfileReport
profile = ProfileReport(df, title='Pandas Profiling Report')
```

#### 2.2.4 Variable Distribution

The Exploratory Data Analysis (EDA) in subsequent stages, through continuous iterations, will provide a clearer view of the distribution of clinical and demographic variables, such as the distribution of diagnoses, length of stays, and visits. This analysis also seeks to identify patterns and anomalies in the data that may be of interest.

#### 2.2.5 Preprocessing Strategy

Data preprocessing focuses on managing missing data, inconsistencies, and transforming high-cardinality variables. Additionally, it seeks to generate new variables (feature engineering) that can enrich subsequent analyses. NLP techniques can be used to extract and categorize relevant information from free-text variables like "Main Diagnosis". For example, adding the year for each data set, the use of different antibiotics, grouping diagnoses, discharges, admissions, etc.

## 1.2 Kedro project Documentation

### 1.2.1 Kedro Project

#### Subpackages

#### Pipelines Documentation

#### hado.pipelines package

#### Subpackages

#### Pipeline Data Preprocessing

#### Submodules

#### hado.pipelines.data_preprocessing.nodes module

#### hado.pipelines.data_preprocessing.pipeline module

#### Pipeline Data Processing

#### Submodules

#### hado.pipelines.data_processing.nodes module

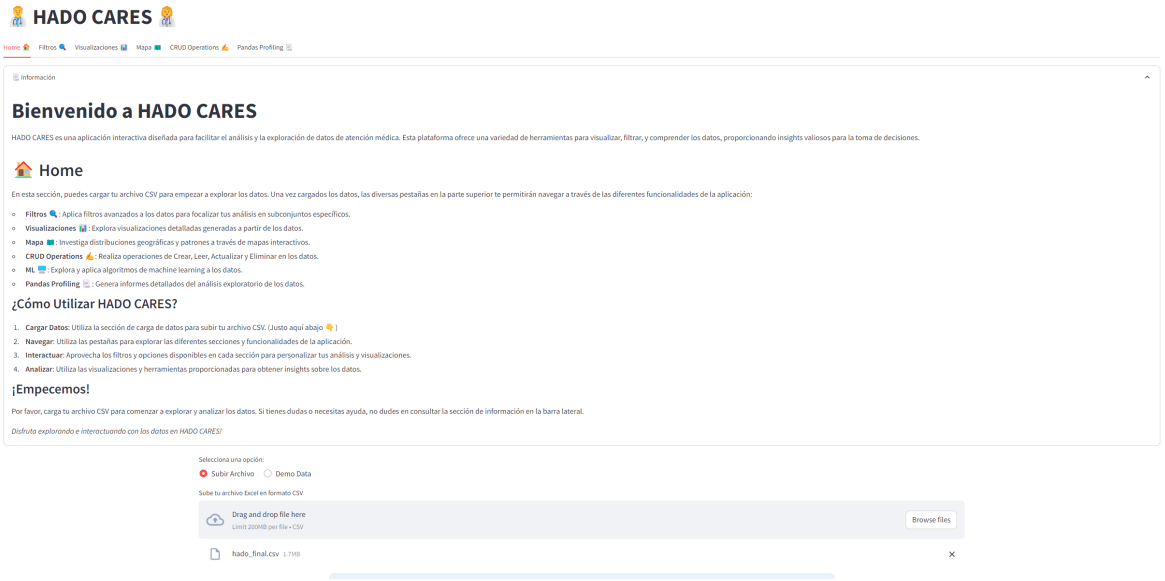
## 1.3 Streamlit app Documentation

### 1.3.1 Streamlit

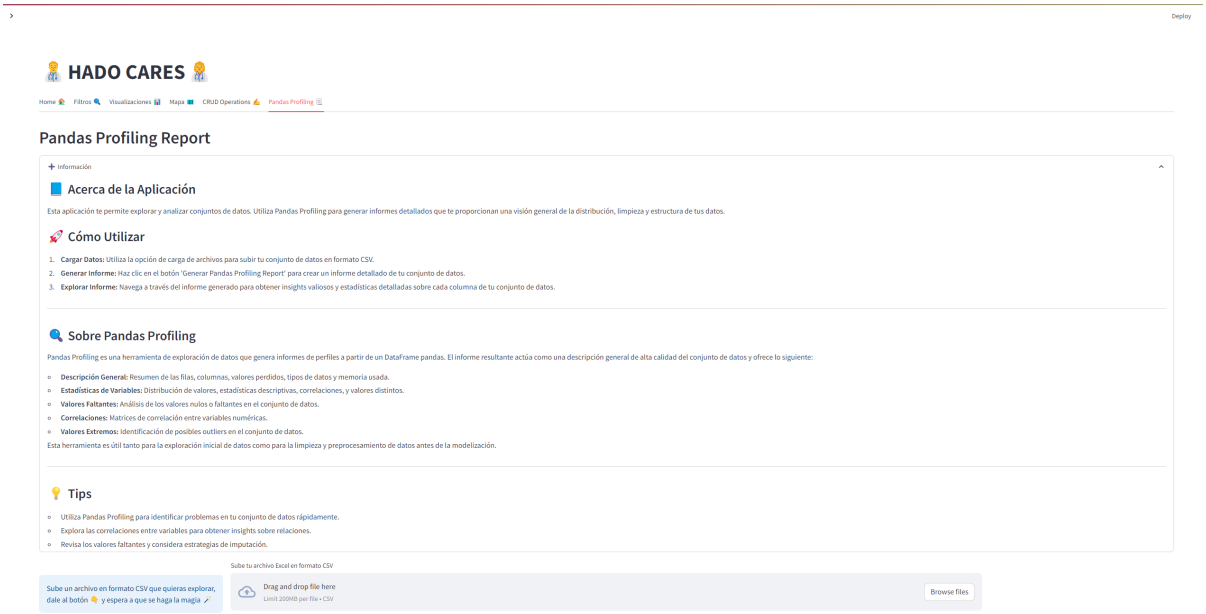
This is the documentation for "HADO_CARES" hado_app

You can access the HADO CARES application here: https://hado-cares.streamlit.app/

- Home:




- Other resources:



### Overview

HADO_CARES is a Streamlit application designed to help HADO department from Spain located at Santiago de Compostela. This documentation will guide you through each module and functionality embedded within the application.

### Prerequisites

Before you get started, you're going to need a few things:

- Your favorite IDE or text editor
- Python 3.8 - Python 3.11
- PIP

  In this case for this app you will need Python 3.10

### Getting Started

To install and run the HADO_CARES Streamlit application, ensure you have met the prerequisites and follow the steps below:

### Step 1: Clone the Repository

Clone the HADO_CARES repository from GitHub to your local machine.

```
git clone https://github.com/pablovdcf/TFM_HADO_Cares.git
```

Navigate to the project directory.

```
cd TFM_HADO_Cares\hado
```

### Step 2: Set Up a Virtual Environment

It's recommended to create a virtual environment to manage dependencies.

```
python -m venv venv
source .\venv\Scripts\activate  # If you are not using Windows use `venv/bin/activate`
```

### Step 3: Install Dependencies

Install the required packages using pip.

```
pip install -r requirements.txt
```

### Step 4: Run the Streamlit Application

Once the dependencies are installed, you can run the Streamlit application.

```
streamlit run hado_app/app.py
```

The application should now be running and accessible in your web browser at *http://localhost:8501*.

### Aditional Notes

- Clone the Repository: Make sure you have Git installed to clone the repository.
- Virtual Environment: Creating a virtual environment is a good practice to manage dependencies in isolation.
- Install Dependencies: requirements.txt should list all the dependencies needed to run the application.
- Run Streamlit: Make sure Streamlit is installed and run the application with the provided command.

### Data Processing Module

The provided suite encompasses a series of functions designed to streamline the data processing and data loading processes within a Streamlit application. Initially, the necessary modules and libraries such as Streamlit, Pandas, ydata_profiling, GeoPandas, and others are imported to ensure the availability of required functionalities.

### Loading Data

- *load_csv_home_expander* allows for the loading of CSV files while providing a summary of the loaded data along with basic statistics and data distribution insights.
- *sidebar_and_upload* handles file upload operations, displaying a sidebar for user interaction and leveraging Streamlit's caching mechanism to optimize performance.

### Data Filtering

- *apply_filters* facilitates interactive data filtering based on various criteria such as year, council, patient status, and others, enabling users to narrow down the data to their specific areas of interest.

### CRUD Operations

- *crud_operations* provides a structured interface for performing Create, Read, Update, and Delete (CRUD) operations on the data, empowering users to not only view but also modify the data interactively.

### Data Profiling

- *generate_pandas_profiling* enables the generation of detailed Pandas Profiling reports from an uploaded file, aiding in the explorative analysis of the data.

### Geographic Data Handling

- *load_gdf* simplifies the loading and cleaning of GeoDataFrames from remote geojson files, ensuring the data is apt for further geographic analysis.

### Additional Notes

Each function is meticulously documented, ensuring clarity on the parameters required and the operations performed, thus promoting ease of understanding and extendibility. The diverse functionalities encapsulated within these functions contribute towards a robust and interactive data processing and management framework within a Streamlit application.

### Data Test Module

This module provides functions to generate synthetic test data that simulates a dataset within a healthcare domain.

### Functions

### Function Details

**generate_data**

The `generate_data` function produces synthetic data with various attributes, aiming to simulate a dataset within a healthcare domain. This function generates random data for a specified number of entries (`n`). The produced data mimics a realistic healthcare dataset, with multiple attributes regarding patient, hospital, and treatment information. NumPy's random choice functionality is used to generate random values for each attribute. The function also defines classifications for certain attributes based on their generated numerical values and encapsulates all the generated data into a Pandas DataFrame, which is then returned.

Here is a brief overview of some of the attributes that are generated:

- **Hospital Information**: Data such as the name of the hospital and service origin.
- **Diagnosis and Admission**: Including categories of diagnosis and reasons for admission and discharge.
- **Geographic Information**: Such as the municipality of origin of the patients.
- **Quantitative Variables**: Such as the number of stays, visits, and evaluative scores.
- **Classifications**: For certain numerical values, classifications are predefined, such as classifications for GDS Fast, Barthel, and PS ECOG scores.

### Example Usage

```
import pandas as pd
import numpy as np

def generate_data(n):
   # Generating random data for each of the numerical columns
   gds_fast = np.random.randint(0, 7, n)
   barthel = np.random.randint(0,100, n)
   ps_ecog = np.random.randint(0, 4, n)

   hospital = np.random.choice(['Santiago', 'Coruña', 'Vigo', 'Ponetvedra', 'Ourense',
↪'Lugo', 'Barbanza'], n)
   servicio_procedencia = np.random.choice(['Unidad Paliativos', 'Oncologia', 'MIR',
↪'Digestivo', 'Urgencias', 'Otros',
                                            'Hematologia', 'Neumologia', 'Cardiologia',
↪'Neurologia'], n)
   diagnostico_categoría = np.random.choice(['Canceres y neoplasias', 'Neurologicas',
↪'Hepaticas y pancreaticas',
                                             'Hematologicas', 'Pulmonares y respiratorias
↪', 'Otros',
                                             'Renales y urinarias', 'Infecciones',
↪'Musculoesqueléticas y de piel',
                                             'Cardiacas'
                                             ], n)
   ingreso_categoría = np.random.choice(['Sintomas', 'Evaluaciones', 'Otros',
↪'Tratamientos'], n)
   atencion_primaria = np.random.choice(['no', 'si'], n)
   numero_estancias = np.random.randint(0, 305, n)
   numero_visitas = np.random.randint(0, 100, n)

   # Defining the corresponding classifications
   gds_fast_classification = [
     ['No realizado o desconocido', 'Deficit cognitivo muy leve', 'Deficit cognitivo leve
↪',
      'Deficit cognitivo moderado', 'Deficit cognitivo moderadamente grave', 'Deficit␣
↪cognitivo grave',
      'Deficit cognitivo muy grave', 'Ausencia de deficit cognitivo'][i] for i in gds_
↪fast]

   barthel_classification = [
     ['Dependencia total', 'Dependencia severa', 'Dependencia moderada', 'Dependencia␣
↪leve o minima', 'Independencia'][
                 0 if i < 20 else 1 if i < 40 else 2 if i < 60 else 3 if i < 90 else 4]␣
↪for i in barthel]

   ps_ecog_classification = [
     ['Totalmente asintomatico', 'Sintomas leves', 'Sintomas moderados',
      'Necesita ayuda para la mayoria de actividades', 'Encamado el 100%'][i] for i in␣
↪ps_ecog]

   # Creating the DataFrame
```

(continues on next page)

(continued from previous page)

```
  data = pd.DataFrame({
    "hospital": hospital,
    "servicio_procedencia": servicio_procedencia,
    "diagnostico_categoría": diagnostico_categoría,
    "ingreso_categoría":ingreso_categoría,
    'atencion_primaria': atencion_primaria,
    'n_estancias': numero_estancias.astype('int64'),
    'n_visitas': numero_visitas.astype('int64'),
    'eva_ing': eva_ing.astype('int64'),
    'ayuntamiento': ayuntamiento,
    'year': year,
    'gds_fast': gds_fast.astype('int64'),
    'gds_fast_classification': gds_fast_classification,
    'barthel': barthel.astype('int64'),
    'barthel_classification': barthel_classification,
    'ps_ecog': ps_ecog.astype('int64'),
    'ps_ecog_classification': ps_ecog_classification,
    })

  return data

# Example usage:
df = generate_data(100)
```

### Interactive Maps Module

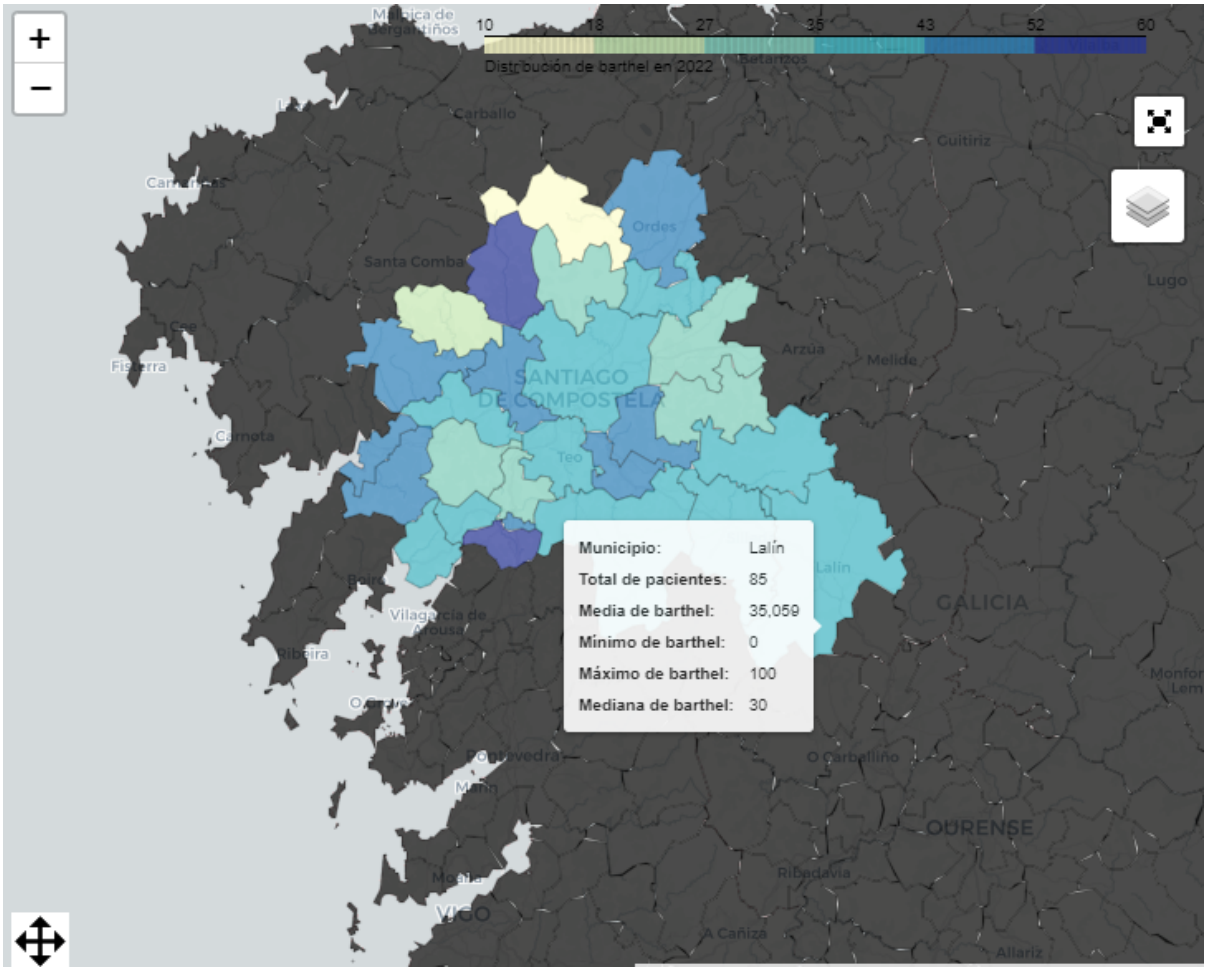
The module under consideration encapsulates a set of functions aimed at rendering interactive maps within a Streamlit application, utilizing data visualization libraries like Folium and data analysis libraries like Pandas. The necessary modules and libraries, including Streamlit, Folium, Matplotlib, and Seaborn, are imported to unlock the functionalities essential for map generation and data visualization.

### Rendering Maps

- *folium_static* takes a Folium map object and renders it within a Streamlit application, ensuring the interactive capabilities of the map are retained in the web app interface.

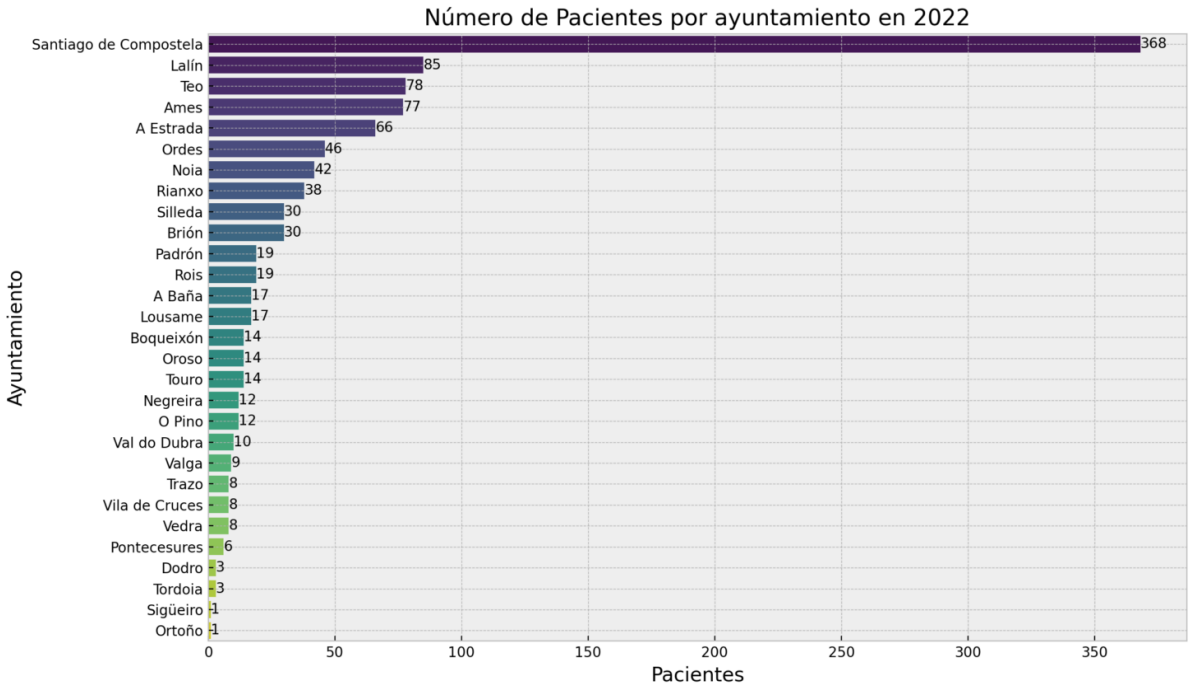
### Interactive Map Generation

- *generate_interactive_maps* dynamically generates interactive maps based on provided data and parameters, offering a visual representation of data distributions across geographical locations.



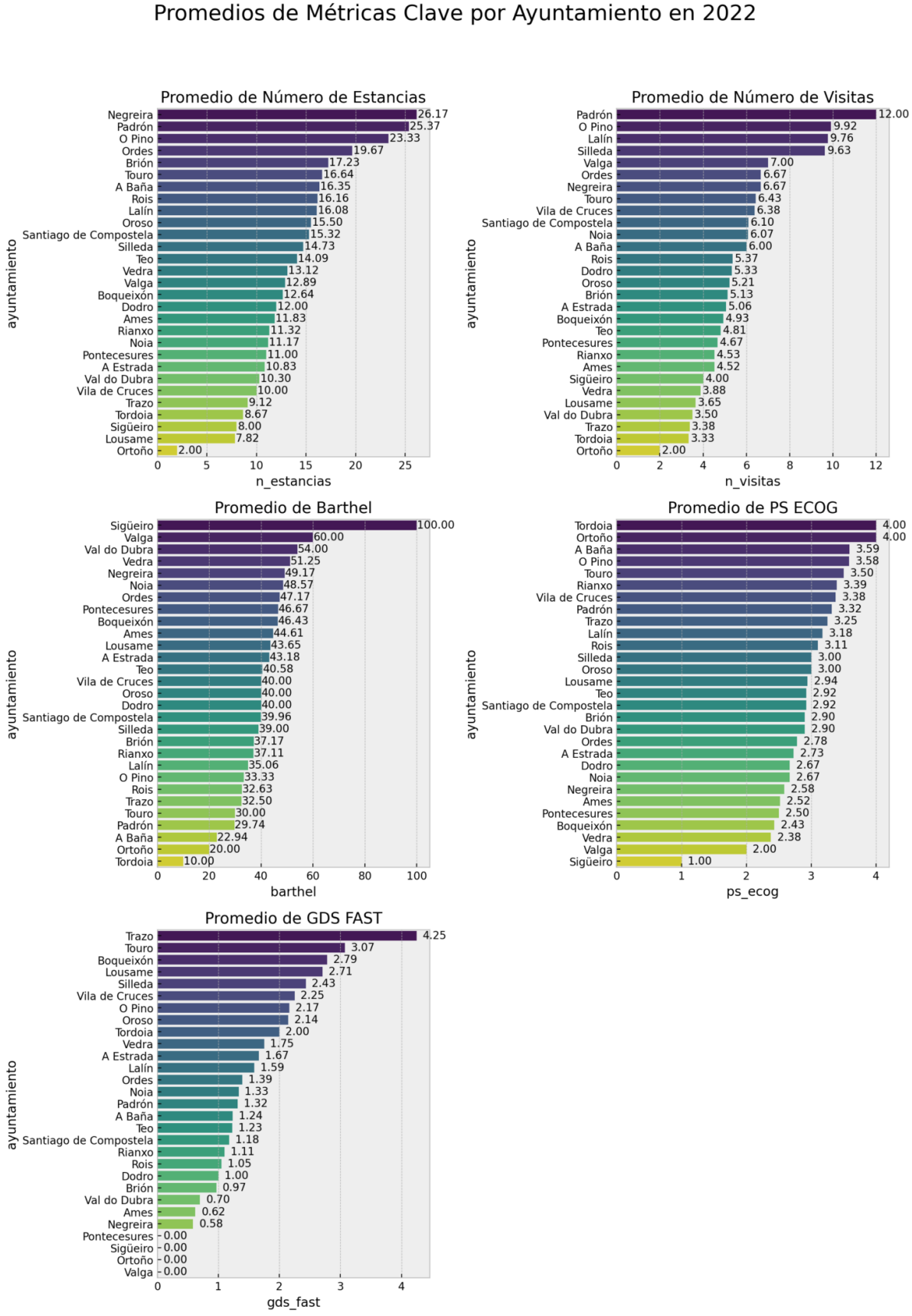

### Visualizing Patient Data by Municipality

- *plot_patients_by_ayuntamiento* visualizes the number of patients per municipality, offering insights into the geographical distribution of patient data.



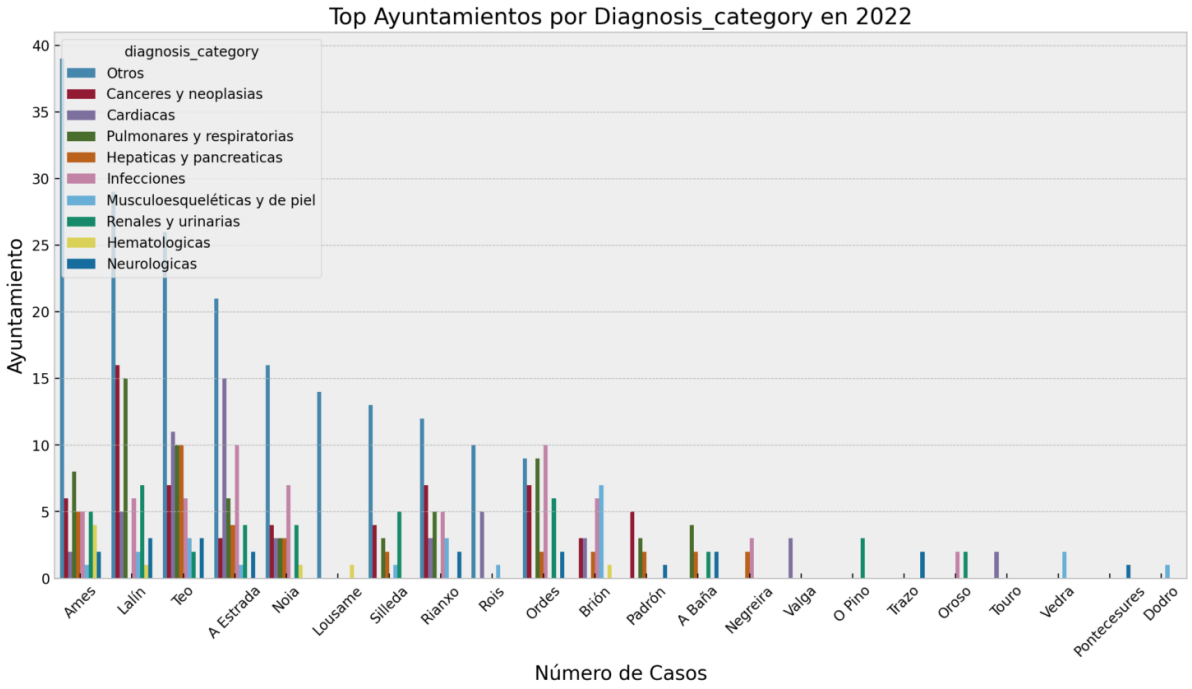

## Visualizing Average Metrics by Municipality

- *plot_average_metrics_by_ayuntamiento* generates visualizations that convey average metrics, providing a bird's-eye view of various key performance indicators across municipalities.

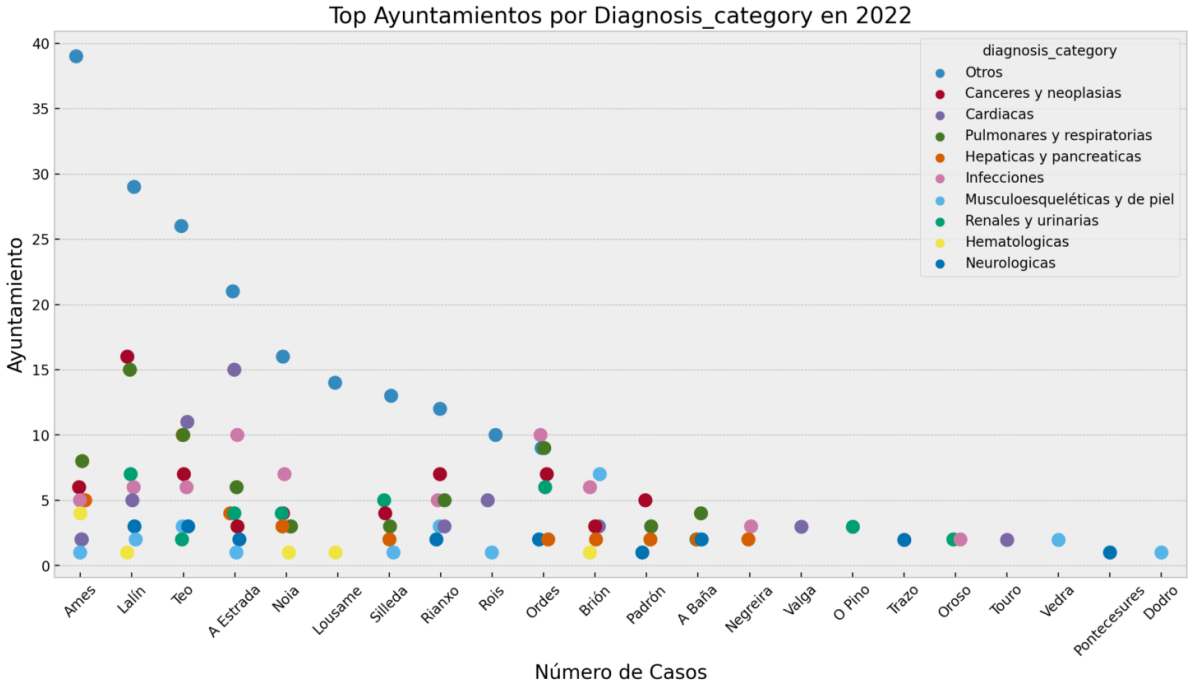

Promedios de Métricas Clave por Ayuntamiento en 2022
Promedio de Número de Estancias
ayuntamiento
n_estancias
Negreira 26.17
Padrón 25.37
O Pino 23.33
Ordes 19.67
Brión 17.23
Touro 16.64
A Baña 16.35
Rois 16.16
Lalín 16.08
Oroso 15.50
Santiago de Compostela 15.32
Silleda 14.73
Teo 14.09
Vedra 13.12
Valga 12.89
Boqueixón 12.64
Dodro 12.00
Ames 11.83
Rianxo 11.32
Noia 11.17
Pontecesures 11.00
A Estrada 10.83
Val do Dubra 10.30
Vila de Cruces 10.00
Trazo 9.12
Tordoia 8.67
Sigüeiro 8.00
Lousame 7.82
Ortoño 2.00
Promedio de Número de Visitas
ayuntamiento
n_visitas
Padrón 12.00
O Pino 9.92
Lalín 9.76
Silleda 9.63
Valga 7.00
Ordes 6.67
Negreira 6.67
Touro 6.43
Vila de Cruces 6.38
Santiago de Compostela 6.10
Noia 6.07
A Baña 6.00
Rois 5.37
Dodro 5.33
Oroso 5.21
Brión 5.13
A Estrada 5.06
Boqueixón 4.93
Teo 4.81
Pontecesures 4.67
Rianxo 4.53
Ames 4.52
Sigüeiro 4.00
Vedra 3.88
Lousame 3.65
Val do Dubra 3.50
Trazo 3.38
Tordoia 3.33
Ortoño 2.00
Promedio de Barthel
ayuntamiento
barthel
Sigüeiro 100.00
Valga 60.00
Val do Dubra 54.00
Vedra 51.25
Negreira 49.17
Noia 48.57
Ordes 47.17
Pontecesures 46.67
Boqueixón 46.43
Ames 44.61
Lousame 43.65
A Estrada 43.18
Teo 40.58
Vila de Cruces 40.00
Oroso 40.00
Dodro 40.00
Santiago de Compostela 39.96
Silleda 39.00
Brión 37.17
Rianxo 37.11
Lalín 35.06
O Pino 33.33
Rois 32.63
Trazo 32.50
Touro 30.00
Padrón 29.74
A Baña 22.94
Ortoño 20.00
Tordoia 10.00
Promedio de PS ECOG
ayuntamiento
ps_ecog
Tordoia 4.00
Ortoño 4.00
A Baña 3.59
O Pino 3.58
Touro 3.50
Rianxo 3.39
Vila de Cruces 3.38
Padrón 3.32
Trazo 3.25
Lalín 3.18
Rois 3.11
Silleda 3.00
Oroso 3.00
Lousame 2.94
Teo 2.92
Santiago de Compostela 2.92
Brión 2.90
Val do Dubra 2.90
Ordes 2.78
A Estrada 2.73
Dodro 2.67
Noia 2.67
Negreira 2.58
Ames 2.52
Pontecesures 2.50
Boqueixón 2.43
Vedra 2.38
Valga 2.00
Sigüeiro 1.00
Promedio de GDS FAST
ayuntamiento
gds_fast
Trazo 4.25
Touro 3.07
Boqueixón 2.79
Lousame 2.71
Silleda 2.43
Vila de Cruces 2.25
O Pino 2.17
Oroso 2.14
Tordoia 2.00
Vedra 1.75
A Estrada 1.67
Lalín 1.59
Ordes 1.39
Noia 1.33
Padrón 1.32
A Baña 1.24
Teo 1.23
Santiago de Compostela 1.18
Rianxo 1.11
Rois 1.05
Dodro 1.00
Brión 0.97
Val do Dubra 0.70
Ames 0.62
Negreira 0.58
Pontecesures 0.00
Sigüeiro 0.00
Ortoño 0.00
Valga 0.00

### Visualizing Top Municipalities for Selected Categories

- *plot_top_ayuntamientos_for_category* visualizes the top-performing municipalities for selected categories, facilitating comparative analysis across geographical locales.
- Bar chart:




- Bubble chart:



### Additional Notes

Each function within the module is documented, providing comprehensive details regarding the expected parameters and the underlying functionality, thereby ensuring that the module can be effectively utilized or expanded upon by other developers. The functions collectively offer a robust framework for generating a variety of interactive visualizations and maps within a Streamlit application, providing a visually intuitive method to explore and interpret geographical data distributions and trends.

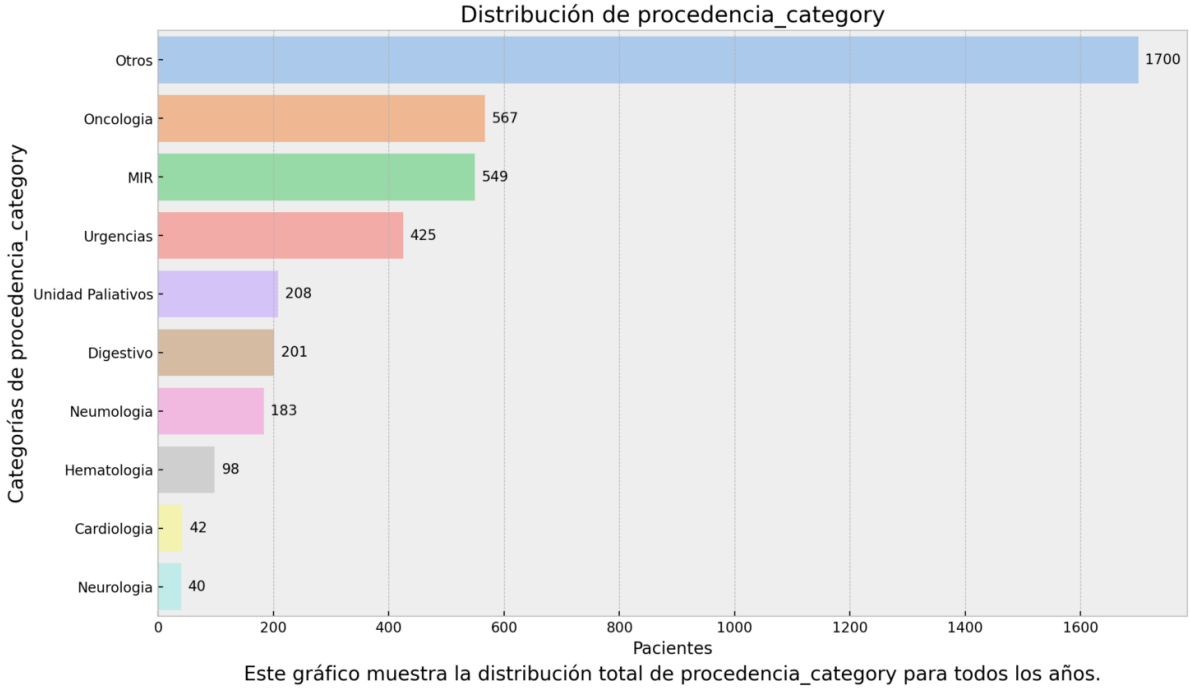
### Visualization Functions Module

This module encompasses a series of functions dedicated to generating various types of visualizations to explore and present the data interactively through a Streamlit application. It utilizes libraries such as Matplotlib, Seaborn, Plotly, and WordCloud to render diverse charts and plots, aiding in the detailed and comprehensive exploratory data analysis.

### Bar and Line Plots

- ***plot_selected_category***

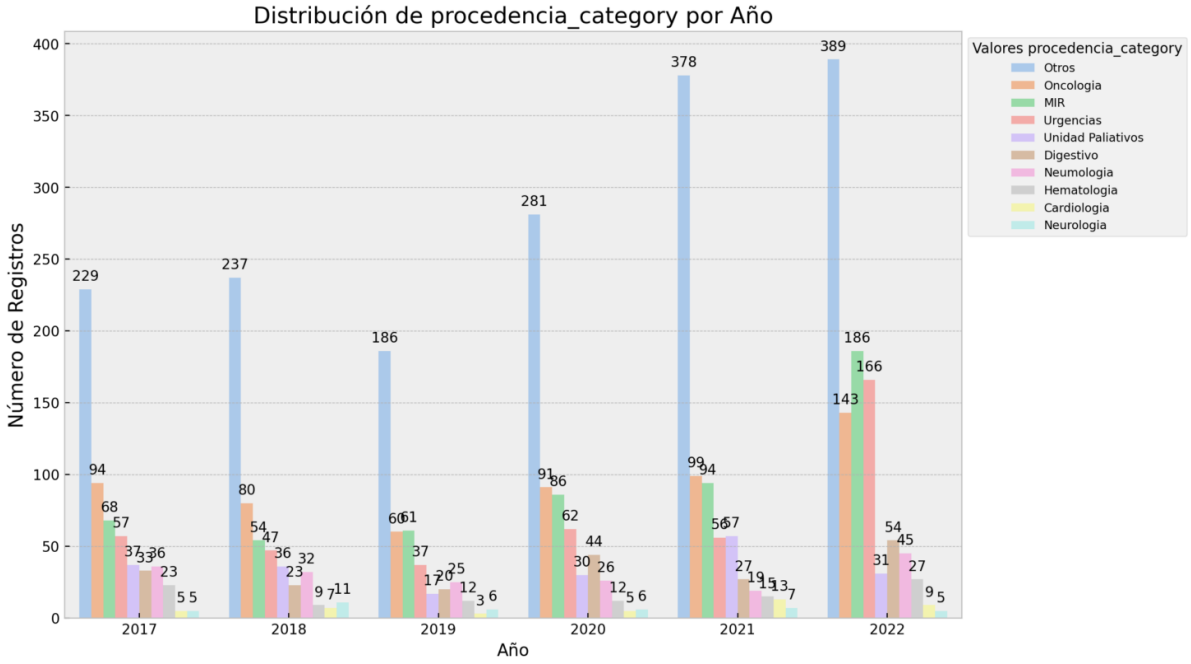
  Generates a bar plot showcasing the distribution of a selected categorical column. Provides insights into the frequency of different categories within a specific column.




Este gráfico muestra la distribución total de procedencia_category para todos los años.

- ***plot_time_trends***

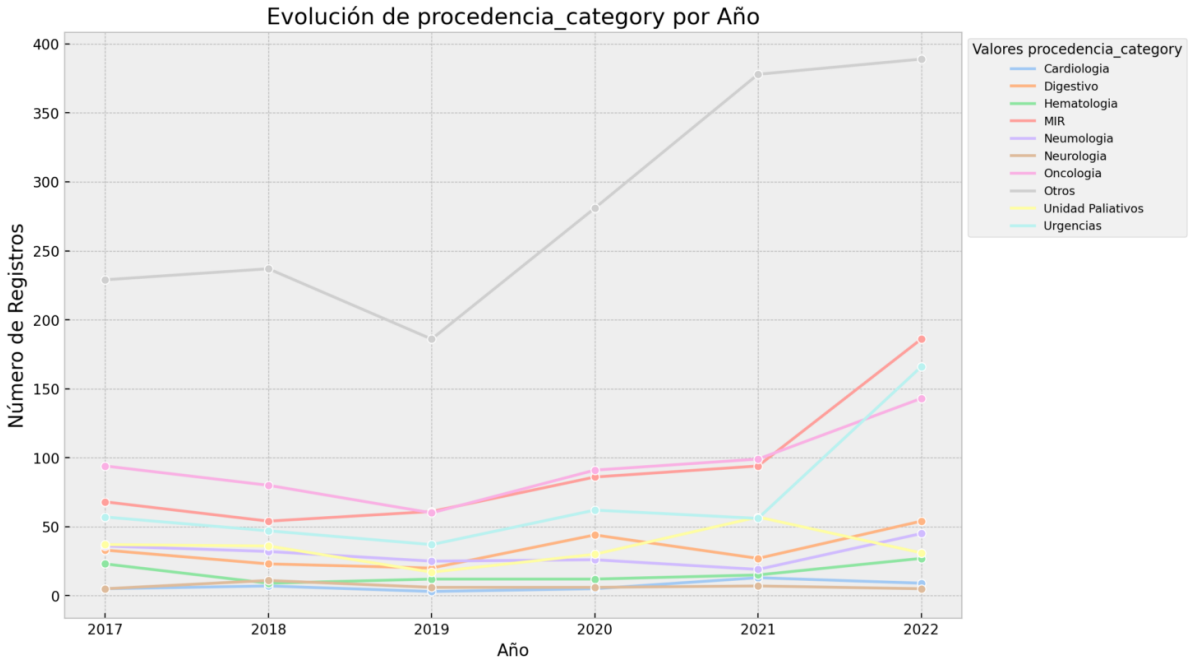
  Visualizes the distribution of a selected categorical column over multiple years, enabling users to perceive trends and variations across different time periods.




Este gráfico muestra la distribución de procedencia_category a lo largo de los años.

- ***plot_time_trends_line***

  Illustrates the evolution of a selected categorical column over the years through a line plot, granting a clear view of the changes and developments across time.

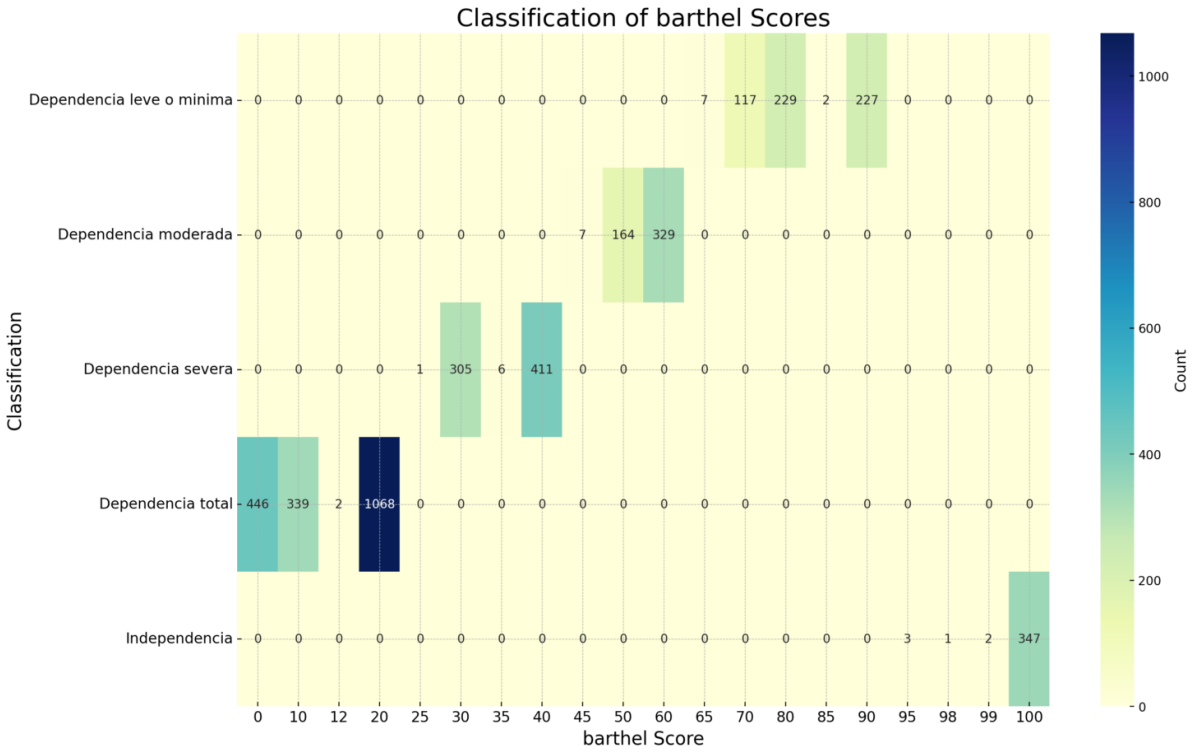



Este gráfico muestra la evolución de procedencia_category a lo largo de los años.

### Heatmaps

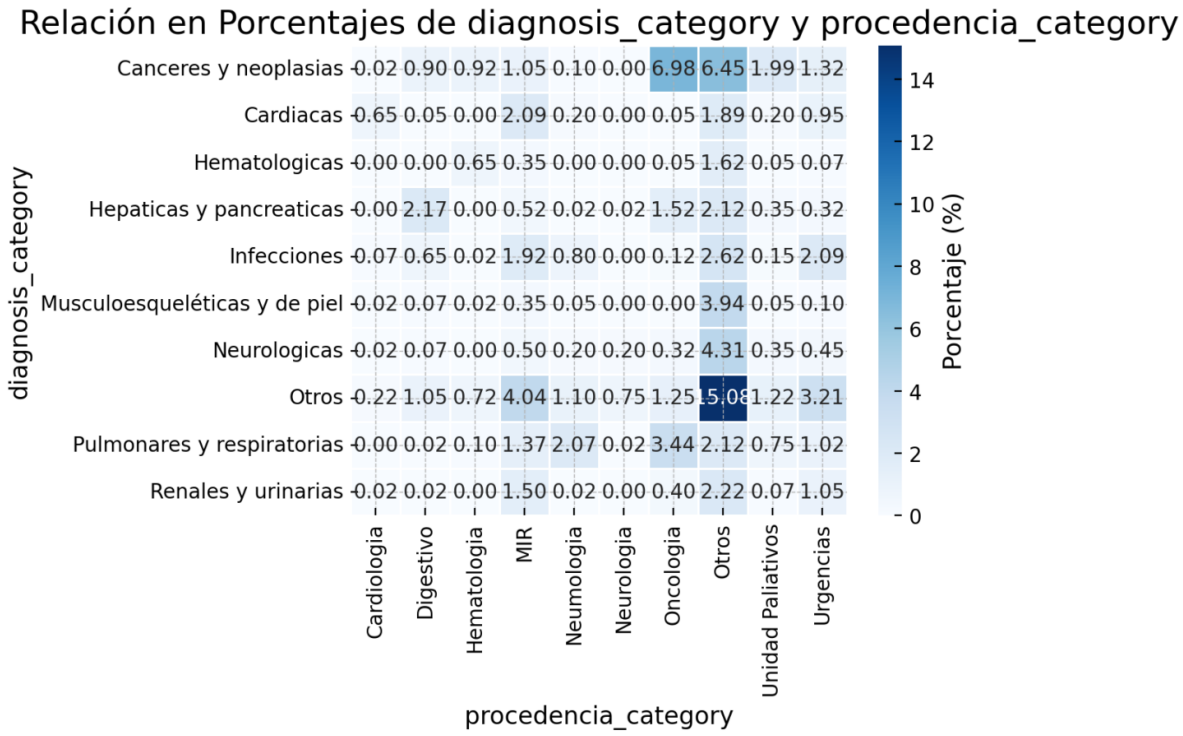
- ***plot_classification_heatmap***

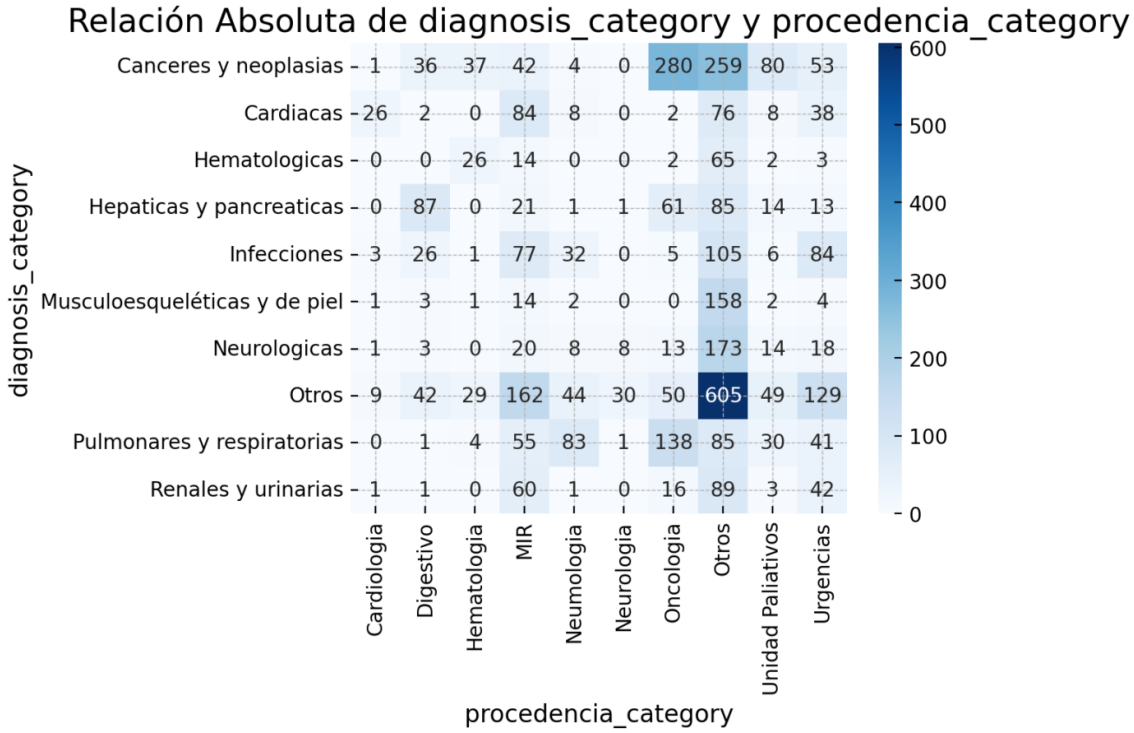
  Presents a heatmap, providing a vivid representation of the distribution of scores for each classification, offering a clear, color-coded visualization of data distributions.




- ***plot_heatmap***

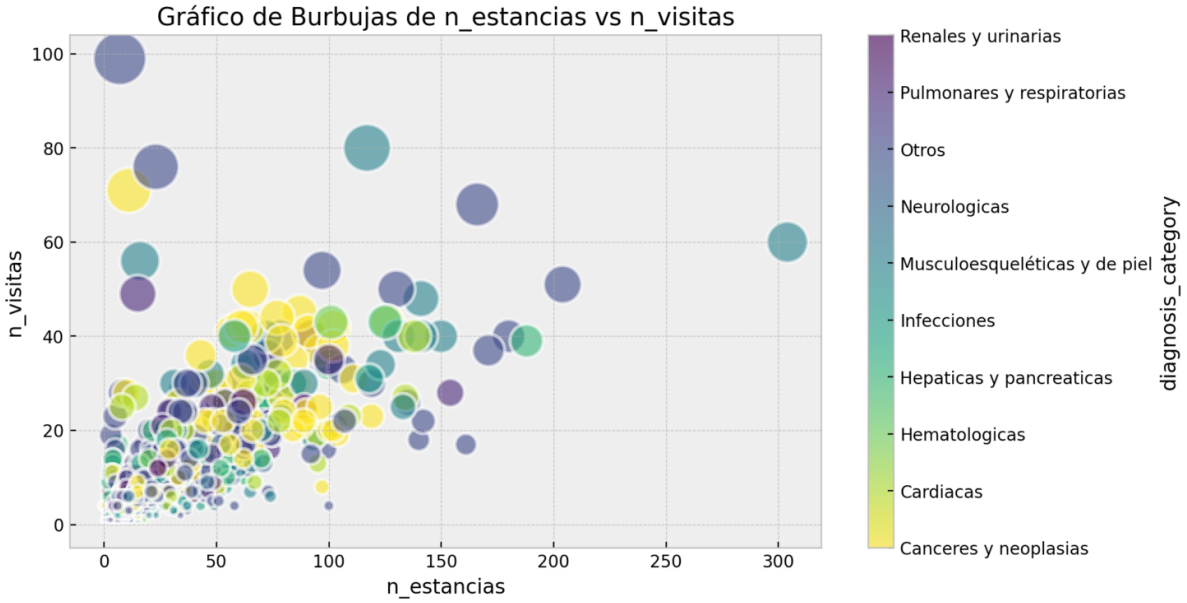
  Deploys two heatmaps displaying relationships between two selected columns, one in percentages and the other in absolute values, providing dual perspectives on the data relationships.


Relación en Porcentajes de diagnosis_category y procedencia_category
diagnosis_category
Canceres y neoplasias 0.02 0.90 0.92 1.05 0.10 0.00 6.98 6.45 1.99 1.32
Cardiacas 0.65 0.05 0.00 2.09 0.20 0.00 0.05 1.89 0.20 0.95
Hematologicas 0.00 0.00 0.65 0.35 0.00 0.00 0.05 1.62 0.05 0.07
Hepaticas y pancreaticas 0.00 2.17 0.00 0.52 0.02 0.02 1.52 2.12 0.35 0.32
Infecciones 0.07 0.65 0.02 1.92 0.80 0.00 0.12 2.62 0.15 2.09
Musculoesqueléticas y de piel 0.02 0.07 0.02 0.35 0.05 0.00 0.00 3.94 0.05 0.10
Neurologicas 0.02 0.07 0.00 0.50 0.20 0.20 0.32 4.31 0.35 0.45
Otros 0.22 1.05 0.72 4.04 1.10 0.75 1.25 15.08 1.22 3.21
Pulmonares y respiratorias 0.00 0.02 0.10 1.37 2.07 0.02 3.44 2.12 0.75 1.02
Renales y urinarias 0.02 0.02 0.00 1.50 0.02 0.00 0.40 2.22 0.07 1.05
Cardiologia
Digestivo
Hematologia
MIR
Neumologia
Neurologia
Oncologia
Otros
Unidad Paliativos
Urgencias
procedencia_category
Porcentaje (%)
0
2
4
6
8
10
12
14


Relación Absoluta de diagnosis_category y procedencia_category
diagnosis_category
Canceres y neoplasias 1 36 37 42 4 0 280 259 80 53
Cardiacas 26 2 0 84 8 0 2 76 8 38
Hematologicas 0 0 26 14 0 0 2 65 2 3
Hepaticas y pancreaticas 0 87 0 21 1 1 61 85 14 13
Infecciones 3 26 1 77 32 0 5 105 6 84
Musculoesqueléticas y de piel 1 3 1 14 2 0 0 158 2 4
Neurologicas 1 3 0 20 8 8 13 173 14 18
Otros 9 42 29 162 44 30 50 605 49 129
Pulmonares y respiratorias 0 1 4 55 83 1 138 85 30 41
Renales y urinarias 1 1 0 60 1 0 16 89 3 42
Cardiologia
Digestivo
Hematologia
MIR
Neumologia
Neurologia
Oncologia
Otros
Unidad Paliativos
Urgencias
procedencia_category
0
100
200
300
400
500
600

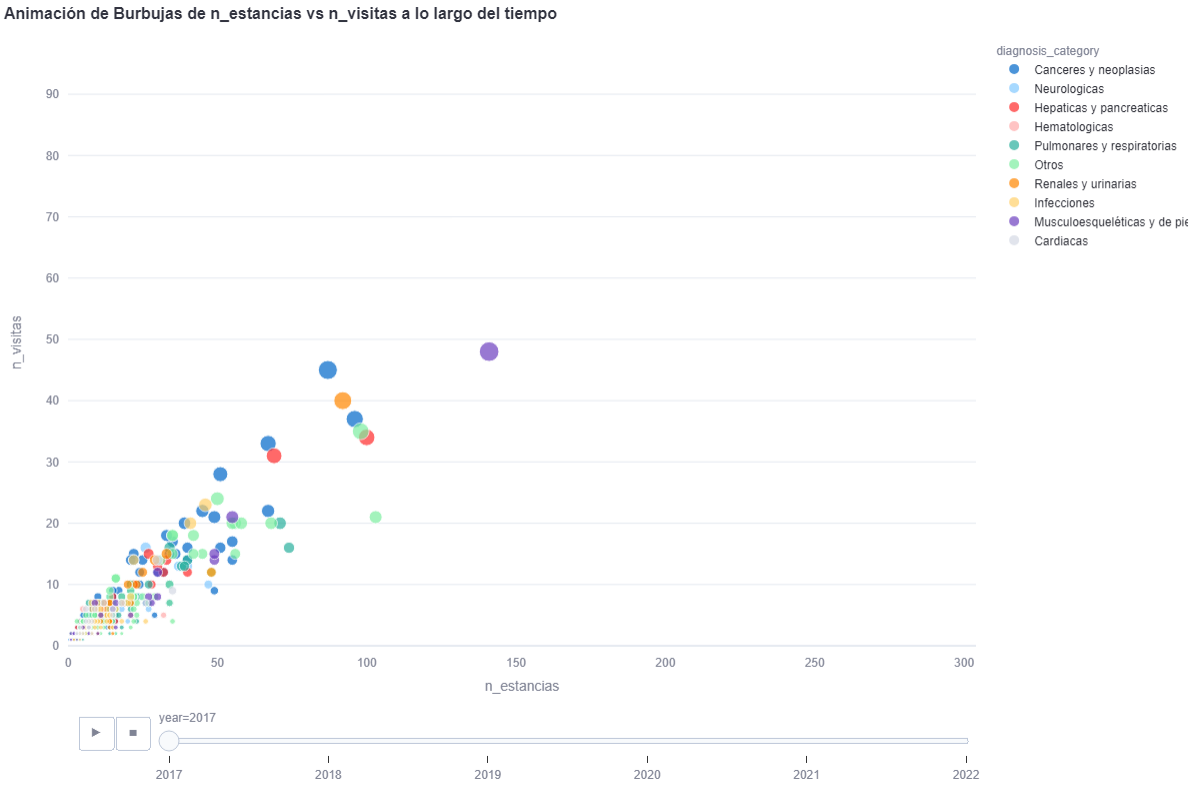
### Bubble Charts

- ***plot_bubble_chart***
  Produces a bubble chart, visualizing the relationship between three or four variables and allowing users to comprehend multi-dimensional data easily.



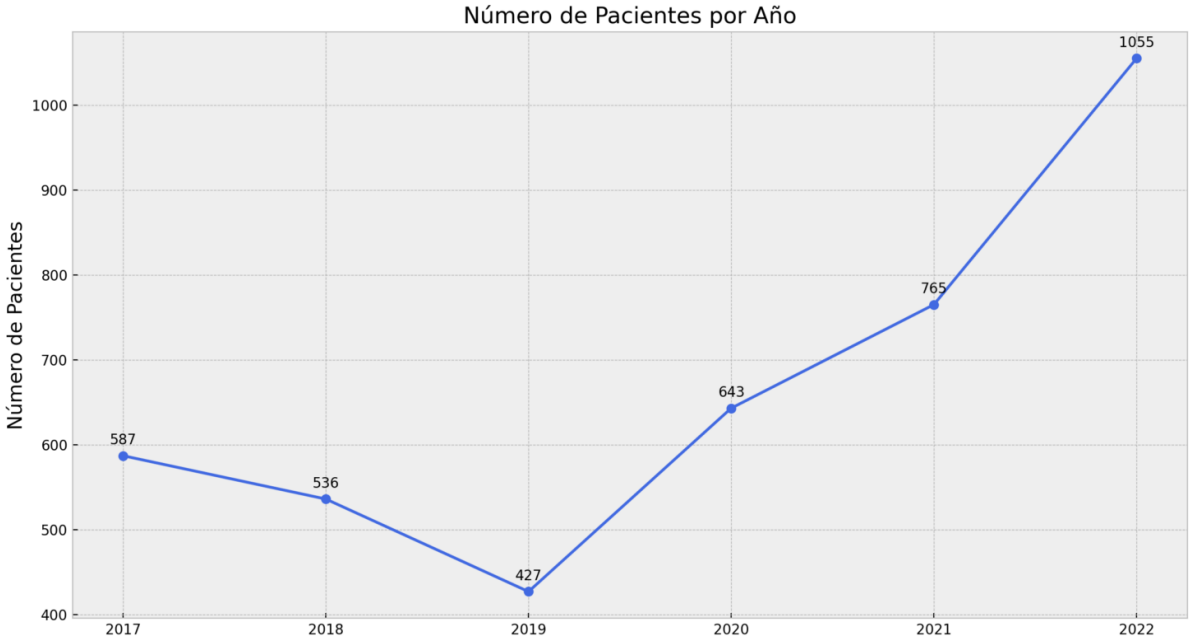

Este gráfico muestra la relación entre n_estancias, n_visitas y n_visitas.

- ***plot_animated_bubble_chart***
  Generates an animated bubble chart that dynamically represents the relationship between three or four variables over time, offering an engaging and intuitive understanding of data evolution.




## Additional Visualizations

- ***plot_total_patients***
  Crafts a line plot depicting the total number of patients per year, providing a straightforward visualization of patient data trends over time.

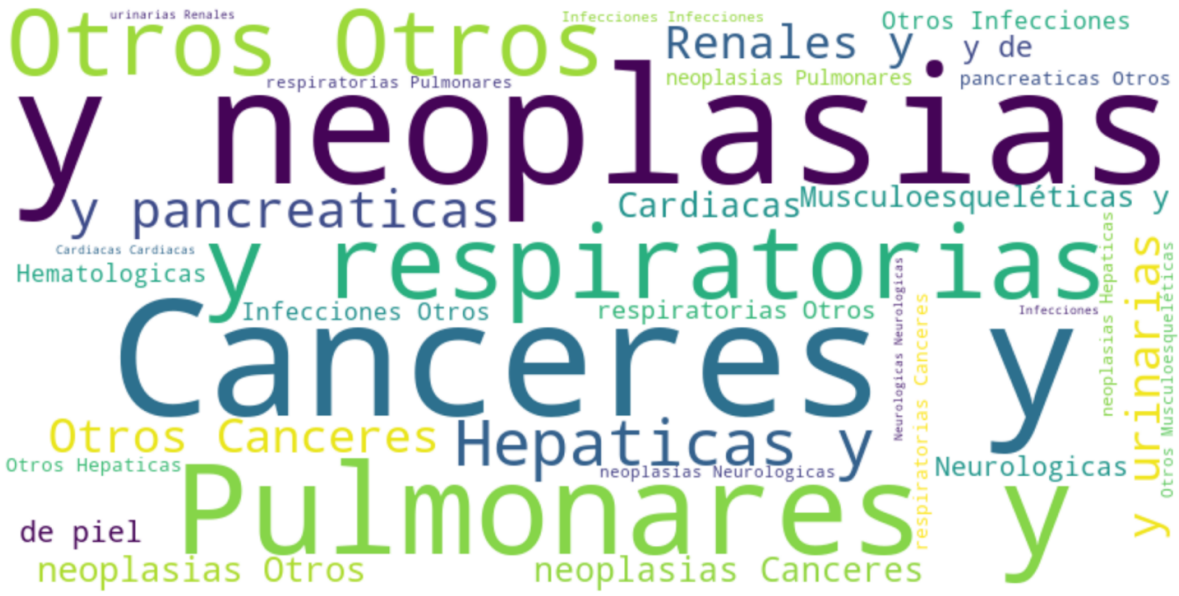



Este gráfico muestra el registro de pacientes a lo largo de los años.

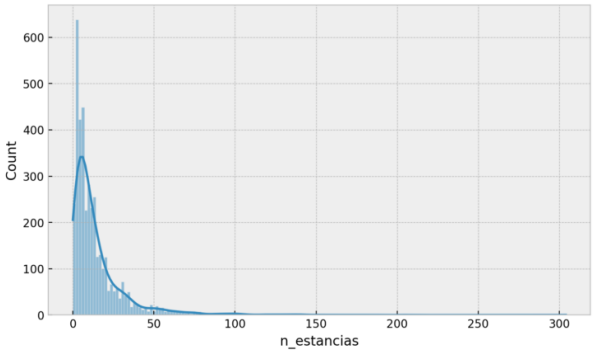
- ***wordcloud_or_hist_box_plot***

  Renders a word cloud for object-type columns or a histogram and a boxplot for numeric columns (int64), offering a flexible method for visualizing both textual and numerical data.

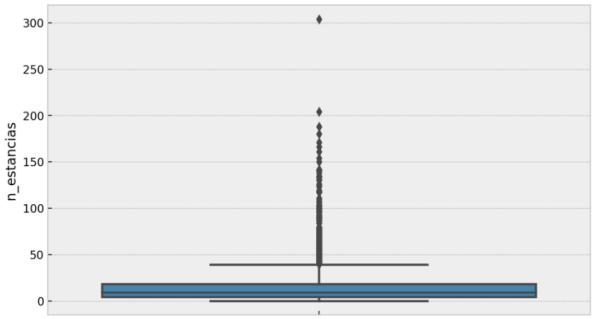
- Wordcloud:




- Histogram and boxplot:




Histograma de n_estancias con densidad de Kernel




Boxplot de n_estancias

### Additional Notes

Each function is meticulously documented, ensuring clarity on the parameters required and the operations performed, thus promoting ease of understanding and extendibility. The functionalities encapsulated within these functions contribute towards a robust and interactive data visualization framework within a Streamlit application.

### Utility Functions Module

The utility functions module contains helper functions designed to improve the user interface and user experience within a Streamlit application. These functions allow developers to add spaces, lines, and informative text to the user interface in a straightforward and efficient manner.

### User Interface Enhancements

- ***ui_spacer***
  This function allows developers to add empty space within the Streamlit application interface. The function takes two optional arguments: *n*, which specifies the number of empty lines to add, and *line*, a boolean that, when true, adds a horizontal line after the empty space.

### Information Display

- ***ui_info***
  This function displays a welcoming message and informational content to the application's main page. It provides a brief description, a gratitude message for users' interest, a disclaimer about the application's testing status, and a link to the source code repository on GitHub.

### Example of Use

```
import streamlit as st
from hado_app.utils import ui_spacer, ui_info

def main():
    ui_info()
    ui_spacer(n=3, line=True)
    st.title("Welcome to HADO Cares Application")
```

### Additional Notes

These utility functions aim to streamline the development of user interfaces within Streamlit applications by providing quick and easy-to-use methods for adding structured informational content and managing spacing within the app. By employing these utilities, developers can ensure a neat and user-friendly interface while maintaining an organized codebase.

CHAPTER

TWO

# INTRODUCTION

In the modern era, where technology is advancing at a rapid pace, the medical field is overwhelmed with an immense amount of data. Although this data holds the potential to unveil valuable insights and enhance patient care, its efficient management and analysis pose a significant challenge, especially for healthcare institutions that still rely heavily on manual and traditional processes for data management.

Specifically, in the HADO area of the Santiago de Compostela hospital, patient records are managed, among other ways, manually through Excel spreadsheets. While functional, this methodology leads to a lack of standardization in data formats and limited utilization of the gathered information, further hampered by the absence of an efficient system to process and analyze the data in a comprehensive and cohesive manner.

"TFM_HADO_Cares", the core project of this Master's Thesis (TFM), emerges in response to this notable gap. It aims to implement a more sophisticated technological and analytical approach with the goal of maximizing the value derived from the collected data, providing deeper insights into diagnoses and treatments, and identifying trends that may be crucial for the continuous improvement of patient care.

The primary objective of this project is to enhance the current patient monitoring process in HADO by: - Conducting exploratory data analysis (EDA) with a special focus on the main diagnoses. - Identifying trends and classifying high-cardinality variables. - Applying Natural Language Processing (NLP) and modeling techniques to group and classify variables. - Creating an application for the visualization of transformed and analyzed data, thus assisting in the decision-making process of HADO professionals.

This TFM will be developed using the Kedro framework to ensure a reproducible and robust data science workflow, and the Streamlit visualization tool to develop an application serving as an interface for data visualization and results. The complete development process, from data collection and preprocessing to analysis and result visualization, will be detailed throughout this document. The code for this project is accessible at the repository: TFM_HADO_Cares.

CHAPTER

THREE

# DATA PROBLEM DESCRIPTION AND ANALYSIS

## 3.1 Data Problem

During the exploratory analysis and data preprocessing process, various challenges associated with data quality and consistency were identified. Among these, inconsistencies and lack of standardization were especially prevalent. The records, originating from multiple annual files and being manually managed, presented a series of inconsistencies and deficiencies that required detailed cleaning and preprocessing.

The challenge also lay in data quality. The need to carry out extensive cleaning and transformations indicated the existence of problems in the raw data, which could be attenuated in the future through a more standardized and automated data collection and management. Additionally, the presence of null values in various critical variables demanded a cautious management strategy to prevent the introduction of biases or errors in subsequent analyses.

## 3.2 Implemented Solutions and Strategies

To address these issues, several strategies and solutions were implemented. Data standardization was a crucial step, where a unified and coherent dataset was created through the concatenation and standardization of various annual datasets. Additionally, text processing and data handling techniques were used to clean and transform the variables, thus improving the data quality for future analyses. In particular, a strategy was proposed to handle the NA values, grounded on the nature of each variable and the clinical context.

The use of Kedro facilitated a robust and reproducible data science workflow, ensuring that analyses and models could be easily replicated and audited.

## 3.3 Challenges and Future Considerations

Although measures have been taken to mitigate the identified data problems, there are challenges and considerations for the future. Automating the process, from data collection to the initial preprocessing phases, could improve efficiency and consistency in the future. It is also imperative to develop more sophisticated strategies or predictive models to handle missing data, especially in critical variables. Although this work focused on cleaning and preprocessing, subsequent analysis stages should explore the patterns and relationships in the data more deeply, leading to deeper analyses.

CHAPTER

FOUR

# RAW DATA DESCRIPTION

## 4.1 Data Dictionary

The columns for Raw Data:

Table 1: Data Dictionary raw Data

| Column Name | Data Type | Description |
|---|---|---|
| Hospital | Object | Name or identifier of the hospital. |
| Servicio | Object | Service type or department within the hospital. |
| AP | Object | Possibly refers to a specific medical procedure or department. |
| Otros | Object | A category for other miscellaneous entries. |
| Diagnostico | Object | Diagnosis information. |
| Motivo Ing | Object | Reason for admission or inquiry (e.g., symptom control). |
| paliativo Onc | Object | Indicates if palliative care for oncology is provided. |
| Paliativo No Onc | Object | Indicates if palliative care for non-oncology is provided. |
| Fiebre | Object | Indicates presence of fever. |
| Disnea | Object | Indicates presence of shortness of breath. |
| Dolor | Object | Indicates presence of pain. |
| Delirium | Object | Indicates presence of delirium. |
| Astenia | Object | Indicates presence of asthenia (weakness). |
| Anorexia | Object | Indicates presence of anorexia. |
| Otros.1 | Object | Another category for other miscellaneous entries. |
| P terminal | Object | Possibly refers to terminal phase of a condition. |
| Agonía | Object | Indicates presence of agony. |
| PS/IK | Object | Possibly refers to performance status or a specific score/index. |
| Barthel | Object | Possibly refers to the Barthel Index, a measure of disability. |
| GDS-FAST | Object | Possibly refers to the Global Deterioration Scale or Functional Assessment Staging Test. |
| EVA ing | Object | Possibly refers to a type of assessment or score at admission. |
| Otros.2 | Object | Another category for other miscellaneous entries. |
| Complicaciones | Object | Indicates presence of complications. |
| Nº estancias | Float64 | Number of stays or admissions. |
| Nº visitas | Float64/Object | Number of visits. |
| SEDACIÓN | Object | Indicates presence of sedation. |
| Mot. ALTA | Object | Reason for discharge or end of care. |
| Médico | Object | Name or identifier of the medical professional. |
| unnamed.1 | Float64 | Unspecified column with sparse data, likely an error or misplaced data. |

## 4.2 General Data Description

The data subject to this analysis comprises **4,013 records** of patients attended in the HADO area of Santiago de Compostela hospital over a span of 6 years. Each record, equivalent to a hospitalization episode, encapsulates a rich variety of demographic, clinical, and administrative data of the patient, detailed in the Data Dictionary.

**Data Quality**: - **Missing Data**: The variable "Discharge Date" presents a notable issue as it is available only until mid-2022, limiting temporal analyses involving this variable. Managing missing data in this and other variables will be vital to ensure the validity of subsequent analyses. - **Data Consistency**: The evaluation of data consistency will be carried out by analyzing outliers and unexpected values, adjusting detected inconsistencies to ensure the reliability of the results. - **Variable Cardinality**: Variables with high cardinality such as "Main Diagnosis" and "ING Reason" may pose a challenge and require strategies like grouping or category transformation to simplify analyses.

**Variable Distribution**: A detailed exploratory data analysis (EDA) will be carried out in subsequent stages to gain more precise insights on the distribution and inherent trends in clinical and demographic variables, such as age, diagnoses, and length of stay.

**Preprocessing Strategy**: Data preprocessing techniques will be implemented to manage missing data, inconsistencies, and transformation of high cardinality variables. Additionally, new variables (feature engineering) will be generated to provide additional value to subsequent analyses, utilizing, for example, NLP techniques to extract and categorize information from free text variables like "Main Diagnosis".

## 4.3 Data Features

The datasets from the HADO area of the Santiago de Compostela hospital vary year by year not only in terms of the number of entries but also in their structure and quality. Below are the main features of the annual datasets from 2017 to 2022:

- **Inconsistencies in Column Names**: The datasets present variability in the column names, reflecting a lack of standardization in data capture and storage.
- **Presence of Null Values**: Certain columns, such as "Discharge Date" in 2022 and "Complications" in other years, have a significant number of null values, requiring a considered strategy for missing data management.
- **Variability in Columns**: Some years have additional columns or fewer columns compared to other years, highlighting the need to align and reconcile these differences during preprocessing.

## 4.4 Preliminary Data Analysis

An initial exploration of the data reveals the following key features:

- **Categorical Data**: Most variables are of object type, indicating that they are categorical or text. This may require encoding or grouping techniques to facilitate further analysis and modeling.
- **Numerical Data**: A limited number of columns are of numeric type (float64), such as "No. of Visits," "No. of Stays," and potentially some other variables that may require conversion or imputation.
- **Missing Data**: The presence of null values in various columns indicates that it will be crucial to establish strategies for managing these missing data, whether through imputation, deletion, or some other suitable method.

## 4.5 Proposed Strategies for Data Cleaning

Given this preliminary analysis, some initial strategies for data cleaning may include:

- **Standardization of Column Names**: Aligning the column names across all datasets to ensure consistency and facilitate data concatenation.
- **Handling Null Values**: Establishing and applying strategies for managing null values in the dataset, potentially using different techniques for different types of variables.
- **Data Type Conversion**: Ensuring that variables are stored in the most suitable data type and performing conversions when necessary.
- **Text Normalization**: For text variables, applying text normalization and cleaning techniques to facilitate text analysis and natural language processing in later stages.

## 4.6 General Observations:

1. **Categorical Variables**: - Many categorical variables seem to be binary (yes/no) or have a few unique categories, such as 'Hospital,' 'Service,' 'AP,' etc. - Several categories have a significant number of the 'no' category, which may suggest that many patients did not experience certain symptoms or treatments. - Some categorical variables, such as 'Diagnosis' and 'Ing Reason,' have many unique categories that could represent a wide range of conditions or reasons for admission.
2. **Numerical Variables**: - 'No. of Visits' and 'No. of Stays' are numerical variables and seem to have some outliers that could distort the mean. - It's interesting to note that 'No. of Stays' has extremely high values in some datasets, but not in others. This might be worth investigating further to understand whether these values are legitimate or data errors.

## 4.7 Suggested Steps for Analysis:

1. **Data Cleaning**: Identify and handle missing values, verify and correct possible outliers, and review and correct inconsistencies in categories.
2. **Exploratory Data Analysis (EDA)**: Includes visualizing distributions, investigating relationships between variables, and examining trends and patterns.
3. **Variable Transformation**: May include converting categorical variables into dummy variables and normalizing numerical variables.
4. **Statistical Analysis and Modeling**: Choose a suitable model, validate the model, and interpret the results.
5. **Communication of Results**: Create clear visualizations and document the steps and results.

CHAPTER

FIVE

# METHODOLOGY

## 5.1 Techniques and Methodologies

This project will implement a variety of **machine learning techniques** and **statistical methodologies** tailored to the character and challenges present in the data. Given the nature of the available variables and the project's objectives, **clustering and classification techniques** for high cardinality variables and **predictive models** to identify trends and correlations in the data are anticipated.

## 5.2 Justification of Chosen Techniques

The choice of these techniques is based on the nature of the data and the project objectives. The use of **classification and clustering techniques** is justified by the need to simplify and categorize high cardinality variables, while **predictive models** will be used to explore and understand the correlations and trends present in the data, providing valuable insights that could be utilized to improve patient management.

## 5.3 Data Science Workflow

The data science workflow for this project will be as follows:

1. **Data Preprocessing**: Will include data cleaning, missing value management, and variable transformation.
2. **Exploratory Data Analysis (EDA)**: Variables will be explored to understand their distribution and relationship with other variables.
3. **Feature Engineering**: New variables useful for machine learning models will be generated.
4. **Modeling**: Machine learning models will be built and evaluated to explore relationships in the data and possibly predict variables of interest.
5. **Data Visualization**: Various visualization tools will be used to explore and present the results of the EDA and modeling in a clear and comprehensible manner.
6. **Interpretation and Communication of Results**: Results will be communicated through visualizations and explanatory texts, ensuring they are accessible and understandable to all stakeholders.