Here’s a detailed step-by-step process to help you complete your EDA project while integrating standout techniques:

**Step 1: Define the Project Scope**

1. **Choose a Dataset**
   * Find a dataset relevant to industries or problems that interest you. Some options:
     + Kaggle Datasets.
     + [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/index.php).
     + Google Dataset Search.
   * **Standout Tip:** Pick a dataset that isn’t overused (e.g., avoid Titanic or Iris) to differentiate yourself.
2. **Define Objectives**
   * Create a clear goal for your project. Examples:
     + "Analyze customer purchasing behavior over time."
     + "Identify key factors affecting movie ratings."
   * Break this into **3-5 specific questions**. Example:
     + "Which months have the highest sales?"
     + "Is there a correlation between user reviews and ratings?"
3. **Create a Plan**
   * Draft a flowchart or bullet-point outline of your process:
     + Data cleaning → Exploration → Visualization → Insights → Documentation.

**Step 2: Set Up Your Workspace**

1. **Organize Files**
   * Set up a structured folder system:

rust

CopiarEditar

EDA\_Project/

├── data/ (for raw and processed datasets)

├── notebooks/ (Jupyter Notebook for your analysis)

├── output/ (charts and visuals)

└── README.md (project overview)

1. **Install Libraries**
   * Install necessary libraries:

bash

CopiarEditar

pip install pandas numpy matplotlib seaborn plotly

1. **Initialize Version Control**
   * Use GitHub to track your progress and make your project accessible.

**Step 3: Data Cleaning and Preparation**

1. **Load and Inspect Data**
   * Start with a Jupyter Notebook:

python

CopiarEditar

import pandas as pd

data = pd.read\_csv('your\_dataset.csv')

data.head()

data.info()

data.describe()

* + Note missing values, data types, and potential outliers.

1. **Clean the Data**
   * Handle missing values:
     + Fill with mean/median for numerical data or mode for categorical.
     + Drop if the column/row is not relevant.

python

CopiarEditar

data['column\_name'].fillna(data['column\_name'].mean(), inplace=True)

* + Standardize formats (e.g., dates or currency).
  + Create derived columns (e.g., month/year from a date column).

1. **Document Cleaning Decisions**
   * Record each cleaning step with comments in your notebook for clarity.
   * **Standout Tip:** Explain the business reason behind your cleaning choices (e.g., "Removing rows with missing revenue data ensures accurate financial analysis").

**Step 4: Exploratory Data Analysis**

1. **Univariate Analysis**
   * Examine each variable individually:

python

CopiarEditar

data['numeric\_column'].hist(bins=20)

data['categorical\_column'].value\_counts().plot(kind='bar')

* + Look for distributions, outliers, and central tendencies.

1. **Bivariate and Multivariate Analysis**
   * Study relationships:
     + Correlations for numerical columns:

python

CopiarEditar

import seaborn as sns

sns.heatmap(data.corr(), annot=True)

* + - Boxplots for categorical vs numerical:

python

CopiarEditar

sns.boxplot(x='category\_column', y='numeric\_column', data=data)

* + Create scatterplots or pairplots for potential trends.

1. **Identify Key Insights**
   * Look for patterns, anomalies, or interesting relationships.
   * Example: "Revenue peaks in December, driven by higher product sales in category X."
   * **Standout Tip:** Annotate findings directly on your charts or in comments.

**Step 5: Create Visualizations**

* Design 3-5 professional charts with actionable insights:
  + Line charts for trends.
  + Bar charts for comparisons.
  + Heatmaps for correlations.
* Use **Tableau** for interactive dashboards. Export key findings from Python and upload to Tableau.

**Step 6: Document Your Work**

1. **Summarize Insights**
   * Write a narrative around your findings:
     + Start with the main questions.
     + Provide key answers backed by visuals.
     + Highlight potential actions based on insights.
2. **Create a ReadMe File**
   * Include:
     + Project overview.
     + Dataset description.
     + Key insights.
     + Tools and libraries used.
3. **Export Cleaned Data**
   * Save the processed dataset for further use:

python

CopiarEditar

data.to\_csv('processed\_data.csv', index=False)

**Step 7: Prepare for Visualization Tools (Future Step)**

Since you’re deciding between Tableau and Power BI, export key charts or data summaries for now. When you decide, we can create impactful dashboards tailored for that tool.

**Step 8: Publish the Project**

1. **On GitHub**
   * Include:
     + Clean Jupyter Notebook.
     + Processed datasets.
     + Final visuals and dashboards.
     + A professional ReadMe file.
2. **On Tableau Public**
   * Share your dashboard link and embed it in your ReadMe or portfolio.

Would you like help finding a dataset, formulating questions, or starting any specific step?

1. Por hacer MAÑANA:

* Revisar “data\_cleaning” notebook y crear un texto explicativo para agregar al readme. Guiarme con lo que ya agregué desde chatgpt. Texto data\_importation está listo. PENDIENTE TEXTO DE “DATA\_CLEANING”.
* Ver video de youtube sobre proyecto EDA y empezar el análisis con el flujo de trabajo que se propone ahí. Hacer el track en github y complelentar acciones con lo que chatgpt me ha recomendado hasta ahora.
* Ver como trabajar con columnas especificas para reducir uso de memoria y que al eliminar rows o hacer cambios se concatenen con el original
* QUE HACER CON MISSING VALUES EN “ID”: contarlos, groupby para ver si son nulos sistemáticamente
* Ordernar carpetas para estructurar mejor el proyecto según recomienda chat gpt (ver conversación) (ej. Agregar carpeta outputs) y hacer le push en github
* CHEQUEAR COMO SUBIR DATOS DE NUEVO A KAGGLE Y COMO ORDENAR ESTO EN GITHUB Y GIT. VER QUE NO HAYA PROBLEMAS

## Dataset Structure

The dataset is stored on Kaggle and consists of the following files:

- \*\*Raw Data\*\*

- [Egresos\_Hospitalarios\_2001.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2002.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2003.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2004.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2005.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2006.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2007.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2008.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2009.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2010.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2011.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2012.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2013.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2014.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2015.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2016.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2017.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2018.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2019.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_Hospitalarios\_2020.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- \*\*Processed Data in CSV and parquet format with raw data cleaned and concatenated\*\*

- [Egresos\_2001-2020.parquet](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

- [Egresos\_2001-2020.csv](https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020)

You can download the dataset files from: (https://www.kaggle.com/datasets/mirkopodunavak/egresos-hospitalarios-2001-2020).

**Project overview (PENDIENTE):**

This project focuses on cleaning and preparing the Egresos Hospitalarios dataset for analysis. The raw dataset contains hospital discharge data from 2001 to 2020, and this repository tracks the steps taken to clean, preprocess, and transform the data for analysis.

**DATA IMPORTATION NOTEBOOK**

1. Data Cleaning Overview

The data cleaning process was essential to join all the 20 datasets together, ensuring accuracy and completeness before starting the analysis. The main steps taken during this process are detailed in the Jupyter notebook located in the notebooks folder called "data\_importation". Below is a summary of the key steps:

1. Data Inspection and importation

In this first step, I examined the glossary of definitions (TRACKEAR EN GIT EXCEL FICHA EGRESOS HOSPITALARIOS) for the columns in the datasets (all them share the same columns) to decide which ones are useful for the project purpose. After making that decision I imported the datasets with the relevant columns, change some data types to simplify future analysis and applied transformatios over specific columns to improve the understanding of their values.

1. Handling Missing Data

Considering the datasets are large (at least 1.000.000 records each) and that the joint dataset consists on 26.340.655 rows that represent the total population that have been subject of a hospital discharge, it was chosen to just delete rows that at least contained one column with missing values except on "Patient ID" and "Healthcare facility name”. Around 7% of the IDs were null and in the “data\_cleaning” notebook the missingness of the IDs was analyzed in relation to the rest of the columns before making decision about handling those null values. The null “Healthcare facility names” values were imputed to “unknown” instead of deleting them to avoid losing valuable data from the rest of the columns. Based on the Central Limit Teorem, deleting a relatively small proportion of rows that at least had 1 missing value column from a big sample it is not going to affect the measures of dispersion and central tendency that give a good understanding of the nature of the data and relations to be explored.

Various techniques were applied, including:

* Removing rows with at least one value column with missing data.
* Standardizing Value Columns and Data Types

**DATA\_CLEANING NOTEBOOK:**

1. Customizing data types and column names

Inconsistent column names and data types can cause issues when performing analyses, so I standardized them throughout the dataset. This step includes:

* Renaming columns for clarity and consistency.
* Converting columns to appropriate data types (e.g., date columns to datetime).

For more information, check the specific transformations done in the notebook.

1. Exploring missing Patient IDs:

Overall, the percent of NaN Patient IDs is 7% which is not very high considering that most of them are for newborns or births. So, the decision is about how to handle the missing IDs depending on the nature of the missingness on each column and choose to delete the rows with NaN IDs or impute those values. Before deciding, it was necessary to analyze each result for the variables after grouping the missing IDs with them.

Age: Most of the NaN IDs are for newborns and then the trend is steady and slightly decreases after the age of 60. In this project it is going to be assumed that the correlation between age and being alive especially after 60 years old is negative, so the amount of people with that age is less and consequently the number of NaN IDs.

Type of Missingness: MAR

Healthcare facility name: The proportion of NaN IDs is remarkably higher in some healthcare facilities, but there was not found information about personal data management polices on the website nor via e-mail with the institutions. The NaN IDs in some hospitals were 100% , so for them was not possible to analyze readmissions.

Type of Missingness: MAR

Discharge condition, Health insurance and Sex: As expected mainly alive people have NaN IDs more than death because less people die in a hospital discharge. So there is no evidence to think that there is a specific decision of having NaN IDs considering this variable. The same happens with sex and health insurance: the amount of men and women and Fonasa and Isapre that have NaN IDs is proportional to the valid values, respectively.

Type of Missingness: MCAR

Year: Over time the amount of NaN IDs has been decreasing progressively and probably it is because of improvements in patient data recording and changes in policies that allow better tracing or follow-up of the health history of the patients. To analyze readmissions, it is going to be considered only rows from the year 2008 onwards because between 2007 and 2008 there was a big decrease on missing NaN IDs, so the revisits before 2008 could be underestimated.

Type of Missingness: MAR

Diagnosis: The amount of NaN Patient IDs considering the diagnosis do not show a clear relation with specific diseases or health conditions. In the first place only very rare (not frequent) diagnosis have 100% NaN Patient IDs. Secondly the most frequent and with higher percent of NaN Patient IDs were mainly births and a diagnosis denominated "other physical therapies". However, for the most common births (in general the ones without issues during the procedure) the frequency is high, but the percent of NaN is low. Lastly, the "score" column created out of the max/min normalization shows that there are no diagnoses with an unexpectedly high number AND percent of NaN patient IDs that could highlight an issue related to the diagnosis variable.

Type of Missingness: MAR

Length of stay: central tendency measures for length of stay is similar between NaN and valid IDs; 5,15% of NaN IDs are outliers(long stays) whereas 5,56% of valid ID are anomalies, which is similar for both of the distributions and not unusual to an alarming point; mental health and nutrition issues are consistently associated with both NaN and valid IDs; and the distribution of outliers for NaN and valid IDs is similar. All this shows that the missigness of patient IDs related to the length of the stay is at random (MAR).

Type of Missigness: MAR

General conclusion about missing data

The decision about the missing data is to delete it in all the variables (Sex, Age, Health insurance, Healthcare facility type, Length of stay, Discharge condition, Primary diagnosis code, Primary diagnosis name, Year) without analyzing the missigness of them (which was done in the notebook “data\_importation”) except on "Patient ID" for which the nature of their null values ​​was analyzed, and "Healthcare facility name".

All the variables except of these 2 are related to each other considering the purpose of the project and the core questions, so having missing data in at least one of those variables can lead to bias in the analysis; even after deleting some rows, the dataset still contains a large number of observations (around 26 million), which ensures that the data remains representative of the population. According to the central limit theorem, a sufficiently large sample size allows for reliable statistical inference.

The "Patient ID" missing data was not deleted in the main dataframe (called “df\_final” in the jupyter notebook) because the IDs are not related to the rest of the variables considering the guiding questions of this project in a way that bias could be introduced, as it is shown in an analysis of the NaN values in relation to the rest of the columns. The missingness of the IDs in relation to the other variables was completely at random (MCAR) or at random (MAR) as it was shown above. That, including other aspects that are explained in the jupyter notebook file, contribute to the decision of keeping the rows with NaN ID values in the “df\_cleaned” dataframe, but deleting the column itself to make the file lighter.

However, the NaN IDs were deleted in the dataframe to be used to analyze readmissions (“df\_readmissions”) and just the valid IDs were kept for the analysis of readmissions and patient journeys, that only considers from the year 2008 onwards, due to possible underestimations of readmissions in the previous years because the decrease of NaN IDs is progressive over the years, reaching a steady trend from that year, as stated in the jupyter notebook file.

Aside from this, the "Healthcare facility name" is not a meaningful variable for the project compared to the data that would be lost if the rows with missing data in this column were deleted, but it is useful to identify each hospital to analyze efficiency and other important issues, so the NaN values were imputed to "Unknown".

Various techniques were applied to handle missing data, including:

* Group by each column to analyze missingness of the data in relation to Patient IDs
* Impute missing values in “Healthcare facility name” with a constant “Unknown” to avoid losing valuable data from other columns
* Deleting rows with missing values in “Patient ID” column before exporting readmission analysis dataframe
* Outliers’ detection techniques for variables related to missing Patient IDs, such as interquartile range method (IQR).
* Apply logarithmic transformations to normalize length of stay outliers for missing Patient IDs and visualize them with histograms

For more information, check the specific process done in the notebook.

1. Handling Duplicates

Duplicated rows were identified and removed, except the first occurrence, but all the duplicates were kept if the Patient ID was missing for the general analysis (these were deleted for readmissions analysis) as it is impossible to know if the rows represent a real duplicated (data entry error) or repeated events (different patient but same value columns) with the information available. It was chosen to acknowledge this uncertainty and keep the rows with missing IDs because the data that would be lost from the rest of the columns apart of it is important at an aggregate level of analysis, however the rows were deleted for readmission analysis, because it is necessary to identify the patient. This strategy allows us to keep the data when the goal is to analyze it at an aggregate level (general analysis) and delete it when the focus is on individual patients (readmissions analysis).

Various techniques were applied to handle duplicates, including:

* Keep the first occurrence of a full duplicated value and deleting the rest
* Splitting the data to then concatenate it based on the decision previously explained

The exact steps for identifying and handling duplicates can be found in the notebook.

1. Final Dataset Format and Export

After the cleaning process, 2 datasets were exported for further analysis as a parquet file to ensure more speed to import and process it.

Dataset “df\_cleaned”: includes all the data from the year 2001 to 2020 with patients with and without ID. The data is ready to be analyzed at an aggregate level for the general analysis that explores the main questions in this project.

Dataset “df\_readmissions”: includes the data from the year 2008 to 2020, because the number of patients without IDs was lower than previous years, allowing to make the estimation of readmissions more accurate. This dataset does not include patients with NaN IDs as this is necessary for the analysis.

1. About the notebook and cleaning process

All detailed steps, code, and explanations are available in the Jupyter notebook here: Notebook Link

The notebooks provide a step-by-step breakdown of the actions taken, including code snippets, visualizations, and reasoning behind the decisions made during the cleaning process.

Why the cleaning process was important

The data cleaning steps were critical in preparing the dataset for meaningful analysis depending on the goal, but also to understand the original structure of the data to make every step contribute for a future analysis that is not biased and provides valid and actionable insights for a broad range of stakeholders.

By following these steps, I have ensured that the data is:

* Free from inconsistencies: valid values, correct column names and data types and others
* Ready for exploratory data analysis and further modeling.
* Easily interpretable and usable for stakeholders.

Conclusion

The README serves as a high-level guide to the data cleaning process, and all the detailed steps, including the code used, can be found in the notebooks folder. Feel free to refer to the notebook for a deeper understanding of the cleaning process and how each decision was made.