Tarea 2: Health Landscape – ENO + GRD

IELE756 – Preparación y Análisis de Datos

Leo Ferres, PhD

March 23, 2026

Tarea 2: Health Landscape – ENO + GRD

Points: 10
Released: Thursday, March 26, 2026
Due: Thursday, April 16, 2026 (before class)
Submission: Canvas – PDF export of your notebook + link to your GitHub repo

---

Goal

Build a health portrait of your assigned comunas using two complementary datasets: **ENO** (notifiable-disease surveillance, 2007–2024) and **GRD** (hospital discharge records, 2022–2024). In Tarea 0 you proved you could open and inspect each dataset; in Tarea 1 you built a demographic baseline from the Census. Now you go deeper into the health data: cleaning messy categorical variables, computing rates, building disease profiles by nationality, and producing **comuna-level summary tables** that you will merge with the Census summary in Tarea 3.

By the end of this assignment you should be comfortable with:

- Cleaning real-world categorical data (inconsistent coding, anonymized values, missing categories)
- Computing disease notification rates and hospitalization rates
- Working with ICD-10 codes and diagnostic chapter groupings
- Comparing health profiles between Chilean and foreign-born populations
- Building choropleth maps of health indicators

Part A: ENO – Notifiable Diseases (5 pts)

A.0 Data Loading & Cleaning (1 pt)

A.0.1 Load the ENO dataset

Load `materials/eno/20241218_base_eno_final.csv` (semicolon-delimited, UTF-8-sig encoding). Select only the columns you will need:

```
eno_cols = ["ENO", "anho_notificacion", "region", "codigo_comuna_residencia",
            "nacionalidad", "sexo", "grupo_edad", "nombre_instruccion",
            "cie_10_diagnostico", "diagnostico", "pais_contagio"]

eno = pd.read_csv("materials/eno/20241218_base_eno_final.csv",
                  sep=";", encoding="utf-8-sig", usecols=eno_cols)
```

Report `eno.shape` and `eno.info()`.

A.0.2 Filter to your comunas

The comuna column is `codigo_comuna_residencia`. It contains **numeric codes as strings** (e.g., `"13101"` for Santiago), but a large fraction of rows are anonymized as `"*****"`.

Filter to your assigned comunas:

```
my_comunas_str = [str(c) for c in MY_COMUNAS]
eno_com = eno[eno["codigo_comuna_residencia"].isin(my_comunas_str)]
print(f"Rows in my comunas: {len(eno_com):,}")
```

Important: report how many rows in the full ENO dataset have `codigo_comuna_residencia == "*****"` (anonymized). This is a data limitation you should acknowledge in your analysis. You cannot recover these rows, so your comuna-level counts will be **undercounts** of the true notification burden.

A.0.3 Clean the nationality variable

The `nacionalidad` column has three values: `"Chile"`, `"Extranjero"`, and `"Desconocido"` (unknown). Report the distribution of `nacionalidad` in your filtered data.

For nationality-specific analyses, **exclude** rows where `nacionalidad == "Desconocido"`, but always report how many rows you are dropping. Do not silently discard them.

A.0.4 Report the time span

Show the range of `anho_notificacion` in your filtered data. How many notifications per year do your comunas have? Present as a simple table.

A.1 Notification Trends Over Time (1 pt)

A.1.1 Overall trend

Plot the **total number of notifications per year** in your comunas as a line chart (x-axis: year, y-axis: count). Annotate or comment on any visible jumps or drops (e.g., COVID-19 period effects on surveillance).

A.1.2 Trend by nationality

On the same plot (or a faceted version), break the trend down by `nacionalidad` (Chilean vs. Foreign, excluding "Desconocido"). Comment: has the share of foreign-national notifications changed over time?

A.2 Disease Profiles (1.5 pts)

A.2.1 Top diseases overall

Show the **top 10 notified diseases** (`ENO` column) in your comunas across all years. Present as a horizontal bar chart.

A.2.2 Disease profiles by nationality

For each of the top 5 diseases, compute the **share of Chilean vs. Foreign notifications** (excluding "Desconocido"). Present as a grouped or stacked bar chart.

Comment: are there diseases where foreign nationals are over-represented relative to their share of the local population? Use your Tarea 1 `pct_foreign` to contextualize.

A.2.3 Age-group distribution for the top disease

Pick the single most common disease in your comunas. Plot its **age-group distribution** (`grupo_edad`), split by nationality. Comment on any visible differences in the age profile.

A.3 Spatial View (0.5 pts)

A.3.1 Notification counts by comuna

Compute the total number of ENO notifications per comuna (across all years). Present as a bar chart.

If you have population data from your Tarea 1 summary table, also compute a **crude notification rate** (notifications per 10,000 population) for each comuna. Present as a second bar chart or a small table.

Note: the ENO notification counts are undercounts due to anonymized comunas. State this caveat explicitly.

A.4 Comuna-Level ENO Summary Table (1 pt)

Build a summary table at the **comuna level** with the following columns:

Column	Description
`codigo_comuna`	Numeric comuna code
`nombre_comuna`	Comuna name (from your Tarea 1 data)
`eno_total`	Total ENO notifications (all years)
`eno_chilean`	Notifications where `nacionalidad == "Chile"`
`eno_foreign`	Notifications where `nacionalidad == "Extranjero"`
`eno_desconocido`	Notifications where `nacionalidad == "Desconocido"`
`eno_top3_diseases`	Names of the top 3 diseases (comma-separated)
`eno_rate_per_10k`	Crude notification rate per 10,000 population

For `eno_rate_per_10k`, use `pop_total` from your Tarea 1 summary table as the denominator.

```
eno_summary = eno_com.groupby("codigo_comuna_residencia").apply(build_eno_summary)
eno_summary.to_csv("output/tarea2_eno_summary.csv", index=False)
```

Display the table in your notebook and save it as CSV.

Part B: GRD – Hospital Discharges (5 pts)

B.0 Data Loading & Cleaning (1.5 pts)

B.0.1 Load GRD files for 2022–2024

Load the GRD files for **2022, 2023, and 2024**. Each file is pipe-delimited (`|`), inside a ZIP archive, encoded in **Latin-1**. Select only the columns you need:

```
import zipfile

grd_cols = ["COMUNA", "NACIONALIDAD", "SEXO", "FECHA_NACIMIENTO",
            "FECHA_INGRESO", "FECHAALTA", "TIPOALTA",
            "DIAGNOSTICO1", "DIAGNOSTICO2",
```

```
            "IR_29301_SEVERIDAD", "IR_29301_MORTALIDAD",
            "IR_29301_COD_GRD", "TIPO_INGRESO",
            "ESPECIALIDAD_MEDICA"]

frames = []
for year in [2022, 2023, 2024]:
    zippath = f"materials/grd/GRD_PUBLICO_{year}.zip"
    txtname = f"GRD_PUBLICO_{year}.txt"
    with zipfile.ZipFile(zippath) as z:
        with z.open(txtname) as f:
            df_year = pd.read_csv(f, sep="|", usecols=grd_cols,
                                  encoding="latin-1", low_memory=False)
    df_year["year"] = year
    frames.append(df_year)

grd = pd.concat(frames, ignore_index=True)
print(f"Total discharges (3 years): {len(grd):,}")
```

B.0.2 Filter to your comunas

In GRD, the `COMUNA` column contains **uppercase text names** (e.g., `"PUENTE ALTO"`, `"SANTIAGO"`). You need a mapping from your assigned comuna codes to these uppercase names.

```
my_comuna_names = ["COMUNA1", "COMUNA2", ...]  # your comunas in UPPERCASE
grd_com = grd[grd["COMUNA"].isin(my_comuna_names)]
print(f"Discharges in my comunas: {len(grd_com):,}")
```

Report the number of discharges per year in your comunas.

B.0.3 Compute length of stay

There is no explicit length-of-stay column. Compute it:

```
grd_com["fecha_ingreso_dt"] = pd.to_datetime(grd_com["FECHA_INGRESO"])
grd_com["fecha_alta_dt"] = pd.to_datetime(grd_com["FECHAALTA"])
grd_com["los"] = (grd_com["fecha_alta_dt"] - grd_com["fecha_ingreso_dt"]).dt.days
```

Report the distribution of `los` (mean, median, min, max). Filter out any rows where `los < 0` (data errors) and report how many you removed.

B.0.4 Create a nationality grouping

The `NACIONALIDAD` column contains full country names (e.g., `"CHILE"`, `"PERU"`, `"VENEZUELA (REPUBLICA BOLIVARIANA DE)"`). Create a binary grouping:

```
grd_com["nat_group"] = grd_com["NACIONALIDAD"].apply(
    lambda x: "Chilean" if x == "CHILE" else "Foreign")
```

Report the distribution of `nat_group`.

B.0.5 Join diagnoses to CIE-10

Join `DIAGNOSTICO1` to the ICD-10 lookup table to get human-readable names and diagnostic chapters:

```
cie10 = pd.read_excel("materials/grd/CIE-10.xlsx",
                      sheet_name="CIE 10")

grd_com = grd_com.merge(
    cie10[["Codigo", "Descripcion", "Capitulo"]].drop_duplicates("Codigo"),
    left_on="DIAGNOSTICO1", right_on="Codigo", how="left")
```

Report how many rows failed to match (where `Capitulo` is null after the join).

B.1 Diagnostic Profile (1.5 pts)

B.1.1 Top diagnostic chapters

Group discharges by ICD-10 **chapter** (`Capitulo`). Show the top 10 chapters as a horizontal bar chart. Comment: which broad disease categories dominate hospitalizations in your comunas?

B.1.2 Top specific diagnoses

Show the **top 15 specific diagnoses** (`Descripcion`) as a horizontal bar chart.

B.1.3 Diagnostic chapters by nationality

For the top 5 diagnostic chapters, compute the share of Chilean vs. Foreign discharges. Present as a grouped or stacked bar chart.

Comment: are there chapters where foreign nationals are over-represented? Use your Tarea 1 `pct_foreign` to contextualize (e.g., if foreign-born are 10% of the population but account for 20% of Chapter XV obstetric discharges, that is noteworthy).

B.2 Length of Stay & Severity (1 pt)

B.2.1 Length of stay by nationality

Compute the **mean and median length of stay** (`los`) for Chilean vs. Foreign patients. Present as a small table. Also plot the distribution of `los` (capped at 30 days for readability) as overlapping histograms or box plots, split by nationality.

Comment: are there meaningful differences?

B.2.2 Severity distribution

Using `IR_29301_SEVERIDAD` (0 = no severity, 1 = minor, 2 = moderate, 3 = major), plot the severity distribution as a bar chart, split by nationality.

Consult `materials/grd/TablasMaestrasBasesGRD.xlsx` (sheet "Severidad GRD") for the severity labels.

B.2.3 Discharge type

Using `TIPOALTA`, show the distribution of discharge outcomes (e.g., "DOMICILIO", "FALLECIDO", "DERIVACION"). Compute the **in-hospital mortality rate** (share of discharges where `TIPOALTA == "FALLECIDO"`), split by nationality. Present as a small table.

B.3 Spatial View (0.5 pts)

B.3.1 Choropleth: hospitalization rate by comuna

Using your Tarea 1 population data, compute a **crude hospitalization rate** (total discharges per 10,000 population) for each of your comunas. Create a choropleth map using `geopandas`.

```
import geopandas as gpd

comunas_gdf = gpd.read_file("materials/carto/Comunas/comunas.shp")
# merge your comuna-level hospitalization rates onto the geodataframe
```

The map should include a title, legend, and comuna labels if legible.

B.4 Comuna-Level GRD Summary Table (0.5 pts)

Build a summary table at the **comuna level** with the following columns:

Column	Description
`codigo_comuna`	Numeric comuna code
`nombre_comuna`	Comuna name
`grd_total`	Total discharges (2022–2024)
`grd_chilean`	Discharges, Chilean nationals
`grd_foreign`	Discharges, foreign nationals
`grd_pct_foreign`	% of discharges by foreign nationals
`grd_mean_los`	Mean length of stay (days)
`grd_mean_los_chilean`	Mean length of stay, Chilean
`grd_mean_los_foreign`	Mean length of stay, foreign
`grd_mean_severity`	Mean severity score

Column	Description
`grd_mortality_rate`	In-hospital mortality rate (% FALLECIDO)
`grd_top3_chapters`	Top 3 ICD-10 chapters (comma-separated)
`grd_rate_per_10k`	Crude hospitalization rate per 10,000 population

For `grd_rate_per_10k`, use `pop_total` from your Tarea 1 summary. You will need to map GRD comuna names back to comuna codes to join with your Census data.

```
grd_summary = grd_com.groupby("COMUNA").apply(build_grd_summary)
grd_summary.to_csv("output/tarea2_grd_summary.csv", index=False)
```

Display the table in your notebook and save it as CSV.

Deliverables

Submit on **Canvas** before class on Thursday, April 16:

1. **PDF export** of your Colab notebook (File > Print > Save as PDF, or File > Download > Download .pdf).
2. **Link to your GitHub repository** (the notebook and both summary CSVs should be committed).

Your notebook must include:

- Markdown cells explaining each step and interpreting results
- All code cells executed with visible output
- The following visualizations:
  - ENO notifications over time, by nationality (Part A.1)
  - Top diseases bar chart (Part A.2.1)
  - Disease profile by nationality (Part A.2.2)
  - Age-group distribution for top disease (Part A.2.3)
  - Top diagnostic chapters and top specific diagnoses (Part B.1)
  - Diagnostic chapters by nationality (Part B.1.3)
  - Length of stay distribution by nationality (Part B.2.1)
  - Severity distribution by nationality (Part B.2.2)
  - Choropleth map of hospitalization rate (Part B.3)
- Both comuna-level summary tables (Parts A.4 and B.4), displayed and saved as CSV

Tips and Common Pitfalls

- **ENO anonymization**: roughly 45% of ENO rows have `codigo_comuna_residencia == "*****"`. Your comuna-level counts are undercounts. Always state this caveat. Do not try to impute the missing comunas.
- **ENO nationality = "Desconocido"**: this category is large (~47% of all ENO records). Exclude it from nationality-specific *rates and shares*, but always report how many rows you are dropping and what fraction of your data it represents.
- **GRD encoding**: the GRD text files use **Latin-1** encoding. If you see garbled characters, add `encoding="latin-1"` to `pd.read_csv()`.
- **GRD decimal separator**: the column `IR_29301_PESO` (DRG cost weight) uses a **comma** as the decimal separator. If you use this column, convert with `.str.replace(",", ".").astype(float)`.
- **GRD comuna names**: the `COMUNA` column contains uppercase text names, not numeric codes. You will need to build a mapping between GRD comuna names and the `codigo_comuna` used in the Census. Your Tarea 1 data (which has both `codigo_comuna` and `nombre_comuna`) can help.
- **Length of stay**: compute `los` as `FECHAALTA - FECHA_INGRESO` in days. Some records may have `los == 0` (same-day discharge); these are valid. Records with `los < 0` are data errors.
- **ICD-10 chapters**: use the `Capitulo` column from the CIE-10 lookup table for high-level grouping. There are 22 chapters.
- **Memory**: GRD files are large (~1M rows/year). Always use `usecols` to load only the columns you need, and filter to your comunas immediately after loading.
- **Reuse your Tarea 1 output**: the `pop_total` and `pct_foreign` from your Tarea 1 summary table are needed here for rate computation and contextualization. Load the CSV you saved.

Grading Breakdown

Part	Points
Part A.0: ENO loading & cleaning	1
Part A.1: Notification trends over time	1
Part A.2: Disease profiles	1.5
Part A.3: Spatial view	0.5
Part A.4: Comuna-level ENO summary table	1
Part B.0: GRD loading & cleaning	1.5
Part B.1: Diagnostic profile	1.5
Part B.2: Length of stay & severity	1
Part B.3: Spatial view	0.5

Part	Points
Part B.4: Comuna-level GRD summary table	0.5
Total	**10**

Half of each part's score comes from correct code and output; the other half comes from clear Markdown explanations and thoughtful interpretation of your results.

Typeset with: `pandoc assignments/Tarea2.md -o assignments/Tarea2.pdf --pdf-engine=pdflatex && evince assignments/Tarea2.pdf &`