Tarea 0: Setup & First Contact with All Three Datasets

IELE756 — Preparación y Análisis de Datos

Leo Ferres, PhD

March 6, 2026

Tarea 0: Setup & First Contact with All Three Datasets

Points: 5
Released: Thursday, March 5, 2026
Due: Thursday, March 12, 2026 (before class)
Submission: Canvas — PDF export of your notebook + link to your GitHub repo

---

Goal

Get your tools working, create your team's GitHub repository, and load each of the three course datasets for your assigned region. This is a shallow first contact: you are **not** analyzing anything deeply yet — just proving that you can open, inspect, and filter each dataset.

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

- Google Colab (or a local Python environment)
- Creating and using a GitHub repository
- Reading parquet, semicolon-delimited CSV, and pipe-delimited files inside ZIP archives with pandas
- Filtering a large dataset to a single region
- Basic inspection: `.shape`, `.info()`, `.head()`, `.value_counts()`

Part 0: Setup (1 pt)

0.1 Google Colab

1. Go to `colab.research.google.com` and create a **New Notebook**.
2. In the first code cell, run:

```
import pandas as pd
print("Hello, IELE756!")
print(f"pandas version: {pd.__version__}")
```

3. Add a **Markdown cell** at the top of your notebook with:

```
# Tarea 0 --- IELE756
**Team members**: [Name 1], [Name 2]
**Region**: [Your assigned region]
**Date**: [today's date]
```

Every notebook you submit must include Markdown cells explaining what you are doing and why. **Code without explanation is not analysis.**

0.2 GitHub

1. If you do not have a GitHub account, create one at `github.com`. Use a **professional username**.
2. **One member** of the pair creates a new **public** repository for the team project (suggested name: `iele756-region-XX`, where `XX` is your region number).
3. Add your partner as a collaborator.
4. The repository should have at minimum:
   - A `README.md` with team members, assigned region, and a one-line description of the project.
   - A `notebooks/` folder where you will place your `.ipynb` files.

Include the **link to your repository** when you submit on Canvas.

Part 1: Census 2024 (1.5 pts)

The Census 2024 microdata are stored as **parquet** files. The table you need is `personas_censo2024.parquet`.

1.1 Load

Load the persona table into pandas, selecting only the following columns (to save memory):

```
import pandas as pd

persona = pd.read_parquet(
    "personas_censo2024.parquet",
    columns=["region", "comuna", "sexo", "edad",
             "p27_nacionalidad", "p27_nacionalidad_rec",
             "escolaridad", "sit_fuerza_trabajo"],
)
```

1.2 Inspect

Report the following for the **full national table**:

- `persona.shape` — how many rows and columns?
- `persona.dtypes` — what are the data types?
- `persona.head(10)` — show the first 10 rows.
- `persona.info()` — any null values?

1.3 Filter to your region

Filter the table to your assigned region. Remember: in the Census, `region` is a **numeric code** (e.g., `1` for Tarapacá, `13` for Metropolitana).

```
my_region = persona[persona["region"] == YOUR_REGION_NUMBER]
print(f"Rows in my region: {len(my_region):,}")
```

1.4 First look at nationality

Using your filtered table, show the distribution of `p27_nacionalidad_rec` with `.value_counts()`. Compute the percentage of foreign-born residents in your region:

```
foreign = my_region["p27_nacionalidad_rec"] \
    .value_counts(normalize=True)
print(f"% foreign-born: {foreign.get('Extranjero', 0):.1%}")
```

Watch out for common traps:

- `-99` means **missing data**, not "minus 99".
- `region` is an integer, not a string.
- `comuna` is a numeric code (e.g., `1101`), not a name.

Consult the variable dictionary (`materials/census2024/diccionario_variables_censo2024.xlsx`) to decode any codes you encounter.

Part 2: ENO — Notifiable Diseases (1.5 pts)

The ENO dataset is a **semicolon-delimited CSV** covering mandatory notifiable disease reports from 2007 to 2024.

2.1 Load

```
eno = pd.read_csv(
    "materials/eno/20241218_base_eno_final.csv",
    sep=";", encoding="utf-8-sig")
print(f"Total rows: {len(eno):,}")
print(eno.columns.tolist())
```

2.2 Filter to your region

In ENO, `region` is **text** (e.g., `"Región de Tarapacá"`), not a number. Filter accordingly:

```
eno_region = eno[eno["region"] == "YOUR REGION NAME"]
print(f"Rows in my region: {len(eno_region):,}")
```

2.3 Notifications by year

Count the number of notifications per year using the `anho_notificacion` column:

```
eno_region["anho_notificacion"].value_counts().sort_index()
```

Present the result as a **bar chart** (use `matplotlib` or pandas' built-in `.plot(kind="bar")`).

2.4 Top 5 diseases

Show the 5 most frequently notified diseases in your region:

```
eno_region["ENO"].value_counts().head(5)
```

Present the result as a **horizontal bar chart**.

2.5 Nationality distribution

Show the distribution of `nacionalidad` in your region's ENO data using `.value_counts()`. Note: you will likely see a `"Desconocido"` (unknown) category — report it, but do not drop it silently.

Part 3: GRD — Hospital Discharges (1 pt)

The GRD files are **pipe-delimited** (`|`), compressed in ZIP archives, one file per year. Each file has **129 columns**, so you must use `usecols` to load only what you need.

3.1 Load one year

Pick **one year** (2024 is recommended). Load it as follows:

```
import zipfile

cols = ["COMUNA", "NACIONALIDAD", "SEXO", "DIAGNOSTICO1",
        "FECHA_INGRESO", "FECHAALTA",
        "IR_29301_SEVERIDAD", "IR_29301_COD_GRD"]

with zipfile.ZipFile("materials/grd/GRD_PUBLICO_2024.zip") as z:
    with z.open("GRD_PUBLICO_2024.txt") as f:
        grd = pd.read_csv(f, sep="|", usecols=cols,
                          low_memory=False)

print(f"Total discharges: {len(grd):,}")
```

3.2 Filter to your region

In GRD there is **no `region` column**. Instead, `COMUNA` contains the commune name in **uppercase** (e.g., `"IQUIQUE"`). You need to know which comunas belong to your region and filter with `.isin()`:

```
my_comunas = ["COMUNA1", "COMUNA2", ...]  # your region's comunas
grd_region = grd[grd["COMUNA"].isin(my_comunas)]
print(f"Discharges in my region: {len(grd_region):,}")
```

3.3 Join with CIE-10

The `DIAGNOSTICO1` column contains ICD-10 codes (e.g., `"J18.9"`). To get human-readable names, join with the lookup table:

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

grd_region = grd_region.merge(
    cie10[["Código", "Descripción", "Capítulo"]],
    left_on="DIAGNOSTICO1", right_on="Código",
    how="left")
```

3.4 Top 5 diagnoses

Show the 5 most common diagnoses (by `Descripción`) in your region:

```
grd_region["Descripción"].value_counts().head(5)
```

Present the result as a **horizontal bar chart**.

Deliverables

Submit on **Canvas** before class on Thursday, March 12:

1. **PDF export** of your Colab notebook (File > Print > Save as PDF, or File > Download > Download .pdf).
2. **Link to your GitHub repository**.

Your notebook must include:

- Markdown cells explaining each step
- All code cells executed with visible output
- The charts requested in Parts 2 and 3

Summary of key differences between datasets

Aspect	Census 2024	ENO	GRD
Format	Parquet (columnar)	CSV (`;` delimiter)	TXT (`|` delimiter), zipped
Encoding	UTF-8	UTF-8-sig	latin-1
Region ID	Numeric code (`1`)	Text (`"Región de Tarapacá"`)	No region column; filter by comuna name in UPPERCASE (`"IQUIQUE"`)
Size	~19M rows	~333K rows	~1M rows/year, 129 columns

Different agencies, different decades, different standards. **Harmonizing** these representations is a central skill of this course.

Typeset with: `pandoc assignments/Tarea0.md -o assignments/Tarea0.pdf --pdf-engine=pdflatex && evince assignments/Tarea0.pdf &`