Individual Student Project : Streamlit Dashboard

Mano Joseph Mathew | mano.mathew@efrei.fr

#EFREIDataStories2025

#EFREIParis #DataVisualization #Streamlit #DataStorytelling #OpenData #StudentProjects #DataVisualization #DataViz #DataAnalytics #DataScience #MachineLearning #AI #BigData #DataDriven #Python #Streamlit #DashboardDesign #DataStorytelling #VisualStorytelling #InformationDesign #DataCommunication #InsightDriven #DataCulture #CareerInData #DataProfessionals #LinkedInLearning #TechCommunity #OpenData #PublicData

Project Overview

Design and build a data-storytelling web app using **Streamlit**. Your app should load a real, public dataset, clean and analyze it, and present an interactive **dashboard** that guides users through a clear **narrative** (problem → analysis → insights → implications). The focus is on **storytelling**: each chart, metric, and interaction should support the narrative and answer specific user questions.

Approved data portals (choose one dataset or a small, coherent bundle from one source): - French open data:

https://www.data.gouv.fr/datasets

https://gd4h.ecologie.gouv.fr/en/catalogue

https://data.europa.eu/data/combined?locale=en

Tip: Prefer datasets with reasonable size, a clear domain question, and meaningful time/geo dimensions.

Learning Objectives

By the end of the project, you should be able to: - Frame a data question and turn it into a **story arc** with audiences and takeaways. - **Ingest, clean, and validate** open data; document assumptions and caveats. - Build an **interactive dashboard** in Streamlit with clear UX and performance best practices. - Apply **EDA** and light analytics (group-bys, joins, aggregations, simple models where relevant). - Communicate insights visually using appropriate **charts, maps, and annotations**. - Package and ship a reproducible app with a **README**.

Deliverables

1. **Streamlit app** (deployable) with a coherent narrative and interactive components.
2. **Short demo video** (2–4 minutes) walking through the story and interactions.
3. **Files included inside the zip file**
    - **Dataset / Link used**
    - **Readme**
    - **Python code**
4. **File name : Example**
    - **StreamlitApp25_20000_NOM_BDML1.zip**
    - **StreamlitApp25_20000_NOM_BDML2.zip**

Scope & Suggested Narrative Patterns

Pick **one** pattern (or combine sensibly): - **Before/After change over time** (e.g., air quality or energy mix trends). - **Compare groups/regions** with a map and small multiples. - **Rankings & distribution** (top/bottom N, inequality, outliers, uncertainty). - **Flow/throughput** (sankey, network, or pipeline stages if data supports it). - **What-if exploration** (simple scenario sliders using assumptions you document).

Ensure the narrative has: - **Hook** (why this matters), **Context**, **Key Insight(s)**, **Implications/Next steps**.

Technical Requirements

- **Streamlit ≥ 1.33**.
- Minimum features:
    - Sidebar controls (filters, date range, region, variable selection).
    - ≥ 3 interactive visuals (e.g., line, bar, map) with tooltips/hover.
    - At least one **map** if geo fields exist; otherwise small multiples.
    - A **metrics header** (KPIs) tied to filters.
    - **Data quality section** (missingness, duplicates, validation checks).
- **Performance**: use `st.cache_data`, efficient merges, and pre-aggregation as needed.
- **Accessibility**: meaningful alt text, readable color contrasts, label all axes and units.
- **Reproducibility**: deterministic environment; code runs from a clean checkout.

Evaluation Rubric (100 pts)

1. **Narrative & Problem Framing (25 pts)**
    - Clear audience, questions, and takeaways; storyboard alignment.
2. **Data Work (25 pts)**
    - Sourcing, cleaning, validation, feature engineering.
3. **Visualization & UX (25 pts)**
    - Appropriate chart types, annotations, legends, color choices, interactions.
4. **Engineering Quality (15 pts)**
    - Code structure, caching, performance, reproducibility, documentation.
5. **Communication (10 pts)**
    - Report clarity, demo video effectiveness, transparency about limitations.

Dataset Selection Checklist

- Relevance to a concrete question and audience.
- Sufficient **granularity** (time, region, category) to enable comparisons.
- Clean schema & metadata (data dictionary, units).
- License compatible with academic reuse; cite properly.
- Size manageable locally; or plan for sampling/caching.

Streamlit App Structure (recommended)

```
app.py
  ├─ sections/
  │    ├─ intro.py            # context, objectives, data caveats
  │    ├─ overview.py         # KPIs, high-level trends
  │    ├─ deep_dives.py       # comparisons, distributions, drilldowns
  │    └─ conclusions.py      # insights, implications, next steps
  ├─ utils/
  │    ├─ io.py               # load_data(), fetch_and_cache(), license text
  │    ├─ prep.py             # cleaning, normalization, feature engineering
  │    └─ viz.py              # chart functions to enforce consistent style
  ├─ data/                    # optional cached/processed CSV/parquet
  └─ assets/                  # logos, icons, images
```

Minimal `app.py` skeleton:

```python
import streamlit as st
import pandas as pd
from utils.io import load_data
from utils.prep import make_tables
from utils.viz import line_chart, bar_chart, map_chart
```

```python
st.set_page_config(page_title="Data Storytelling Dashboard", layout="wide")

@st.cache_data(show_spinner=False)
def get_data():
    df_raw = load_data()
    tables = make_tables(df_raw)
    return df_raw, tables

st.title("Data Storytelling: <Your Topic>")
st.caption("Source: <dataset title> — <portal> — <license>")

with st.sidebar:
    st.header("Filters")
    regions = st.multiselect("Region", [])
    date_range = st.date_input("Date range", [])
    metric = st.selectbox("Metric", [])

raw, tables = get_data()

# KPI row
c1, c2, c3 = st.columns(3)
c1.metric("KPI 1", "…", "Δ vs. baseline")
c2.metric("KPI 2", "…")
c3.metric("KPI 3", "…")

st.subheader("Trends over time")
line_chart(tables["timeseries"])  # custom function adds consistent styling

st.subheader("Compare regions")
bar_chart(tables["by_region"])

st.subheader("Map view")
map_chart(tables["geo"])

st.markdown("### Data Quality & Limitations")
st.info("Describe missing data, measurement limits, and biases.")

st.markdown("### Key Insights & Next Steps")
st.success("Summarize what matters and what actions follow.")
```

Visualization & Storytelling Guidelines

- Start with a **headline insight** per section (one sentence).
- Use **annotations** and **reference lines/bands** to highlight thresholds.
- Prefer **small multiples** over over-cluttered legends.
- Label units, axes, and sources; avoid ambiguous colors.

- Add **help text** (`st.help`, tooltips) for methods/definitions.
- Keep interactions purposeful; don't add controls that don't change the story.

Data Quality & Ethics

- Document sampling, known biases, and collection methods from the portal metadata.
- Respect and reproduce **licenses** (code vs data may differ).
- If showing people-related data, aggregate to protect privacy; avoid re-identification risks.
- Be transparent about uncertainty and methods; avoid over-claiming causality.

Reproducibility & Packaging

- Pin dependencies (`requirements.txt`) and test in a clean environment.
- Include a **`Makefile`** or simple **run instructions** in `README.md`.
- Add a **data download script** (with caching) instead of bundling large raw files.
- Provide a `seeds.json` or constants for consistent sampling.

Example `requirements.txt`:

```
streamlit
pandas
numpy
pyarrow
plotly
altair
geopandas  # if using maps
pydeck      # optional map layer
requests
```

Submission

- **URL** to deployed app **after validation** (Streamlit Community Cloud or similar).
- **Repo link** with code and report.
- **Storyboard** + **demo video** link.

Example Themes (inspiration)

- Air quality evolution near low emission zones (GD4H datasets).
- Electricity production mix and CO_2 intensity across EU countries.

- Public transport usage vs. pollution in French metros.
- Water quality compliance by region and season.
- Renewable adoption vs. energy prices.

Grading Hints / What Great Looks Like

- Clear, persuasive narrative with defensible methods and transparent limits.
- Seamless interactions that encourage exploration without confusion.
- Visuals that reveal non-obvious patterns and support decisions.
- Clean, modular code and fast loading.
- Thoughtful discussion of ethics and uncertainty.

Resources

- Streamlit docs: https://docs.streamlit.io/
- Altair: https://altair-viz.github.io/ | Plotly: https://plotly.com/python/
- Geo resources: https://geopandas.org/ | https://deck.gl/
- Storytelling: Evergreen Data, Data Viz Society, FT Visual Vocabulary