**Ministry of Education and Research of the Republic of Moldova**

**Technical University of Moldova**

**Faculty of Computers, Informatics and Microelectronics**

**Discipline: Exploratory Data Analysis**

**REPORT**

**Assignment 1:** European Development Indicators

Performed by:

st. gr. IS-251-M Igor Latii

Verified by:

l. univ. Alexandru Monahov

Chișinău - 2025

**Assignment 1: European Development Indicators**

**Phase 1: Country Selection & Data Acquisition**

1. **Chosen Country**

For this project, the selected European country is Latvia (LV). Latvia represents an interesting case for economic analysis due to its relatively small and open economy, strong dependence on trade and transport flows, and demographic challenges such as migration and population decline. The country’s integration into the EU single market and its adoption of the euro have also made it highly sensitive to regional and global economic dynamics.

1. **Research Questions**

The following research questions (RQ) have been defined to explore the economic relationships and structural dynamics in Latvia using quantitative indicators from Eurostat and the World Bank:

**RQ1**: How has the evolution of external trade and passenger flows correlated with Latvia’s GDP and overall economic activity?

**RQ2**: What is the relationship between unemployment, migration, and international departures of Latvian residents?

**RQ3**: How do freight and passenger transport volumes relate to inflation?

These questions aim to go beyond simple descriptive analysis and investigate the interconnections between trade, mobility, and macroeconomic performance over time.

1. **Selected Indicators (Eurostat / World Bank)**

To address the above research questions, a set of 15 indicators was selected, primarily from Eurostat, with one Net Migration (SM.POP.NETM) sourced from the World Bank via the link: <https://data.worldbank.org/indicator/SM.POP.NETM>, as this measure is not available in Eurostat.

The final selection is documented in [/reports/indicators.csv](indicators.csv) and [/reports/indicators.xlsx](indicators.xlsx) and includes 6 monthly, 2 quarterly, 1 semestrial, and 6 annual datasets.

Mapping of Research Questions to Indicators can be presented as follows:

**RQ Indicators (Code)**

RQ1 namq\_10\_gdp, nama\_10\_exi, avia\_paocc, road\_pa\_mov, sts\_inpr\_m, sts\_trtu\_m, tour\_occ\_nim, nrg\_pc\_202

RQ2 une\_rt\_m, lfsi\_emp\_q, migr\_emi1ctz, SM.POP.NETM, demo\_pjan, tour\_occ\_nim

RQ3 prc\_hicp\_manr, tran\_hv\_frtra, avia\_paocc, road\_pa\_mov, sts\_inpr\_m, sts\_trtu\_m, nrg\_pc\_202

This selection ensures a balanced dataset across economic, demographic, and transport sectors, aligned with the research focus.

1. **Data Sources and Acquisition Approach**

Primary source: **Eurostat** — official statistical database of the European Union, accessed programmatically via the eurostat Python package.

Secondary source: **World Bank Open Data** — used exclusively for the Net Migration (SM.POP.NETM) indicator.

Data collection was automated through a custom Python script [/src/collecting\_data.py](../src/collecting_data.py), which:

1. Reads the list of indicators from [/reports/indicators.csv](indicators.csv).
2. Uses the Eurostat API to download each dataset.
3. Filters observations by the selected country (geo = LV).
4. Saves the resulting tables in [/data/raw/](../data/raw) for further processing.

Each dataset is stored in its raw format (CSV) to ensure full reproducibility and transparency of preprocessing.

The World Bank dataset for Net Migration was downloaded manually in CSV format from (<https://data.worldbank.org/indicator/SM.POP.NETM>).

1. **Data Frequency and Coverage**

The collected datasets vary in temporal frequency:

* Monthly indicators: Inflation, Unemployment rate, Industrial production, Retail trade turnover, Tourism nights spent, Air passenger transport.
* Quarterly indicators: GDP, Employment.
* Semestrial indicator: Energy consumption per capita.
* Annual indicators: Exports and imports, Migration, Freight, Road passenger transport, Population.

The temporal coverage spans from the 1950s to 2025, depending on data availability per series.

Specific data frequency and coverage of each selected indicator is documented in [/reports/indicators.xlsx](indicators.xlsx).

More than half of the indicators (8 out of 15) have monthly or quarterly frequency, satisfying the requirement for high-frequency time series suitable for short-term dynamics analysis.

All data were validated to include only Latvia (geo = LV), ensuring full consistency across datasets and comparability between indicators.

**Phase 2: Data Cleaning and Preparation**

This phase focused on obtaining, cleaning, transforming, and harmonizing the datasets collected for Latvia in order to prepare them for exploratory data analysis (EDA). All operations were implemented in Python using the pandas library, ensuring full reproducibility and transparency of the data preparation workflow.

**1. Data Acquisition and Organization**

The datasets collected in Phase 1 (15 indicators from Eurostat and one from the World Bank) were organized under the directory [/data/raw/](../data/raw).

Each dataset was stored in its original CSV format. To enable reproducible processing, all transformation scripts were implemented as modular Python programs located in [/src/](../src), with results saved to structured subdirectories under [/data/processed/](../data/processed).

Data were processed in **3 main steps**:

**Step 1:** Transformation to Long Format

Eurostat datasets were downloaded in a wide format, where each column represented a time period (e.g., 2010, 2011, …).

Using a dedicated script ([/src/transform\_to\_long\_format\_EStat.py](../src/transform_to_long_format_EStat.py)), the data were converted into a long format with standardized columns:

* TIME\_PERIOD – representing the temporal dimension,
* VALUE – numeric indicator values,
* metadata fields (e.g., geo, unit, na\_item, freq, etc.).

The same logic was applied to the World Bank dataset ([/src/transform\_to\_long\_format\_WB.py](../src/transform_to_long_format_WB.py)), which requires skipping the initial metadata rows and reshaping the time series using the pandas.melt() function.

**Step 2:** Filtering and Country Selection

All datasets were filtered to include only records where geo = LV (Latvia) or Country Name = Latvia for the World Bank file.

This ensured complete consistency across indicators, enabling reliable comparison and merging at later stages.

**Step 3:** Intermediate Storage

The transformed “long” tables were saved to [/data/processed/transformed\_to\_long\_format/](../data/processed/transformed_to_long_format) for further harmonization.

**2. Data Cleaning Procedures**

***a. Handling Missing and Zero Values***

Rows with missing (NaN) or zero values in the VALUE column were treated as absent observations rather than valid data.

Missing values: Automatically removed during transformation using dropna(subset=["VALUE"]).

Zero values: Removed in a subsequent cleaning step ([/src/format\_time\_periods.py](../src/format_time_periods.py)), as zeros in our transport or trade series typically represent unreported or non-existent measurements rather than true zero activity.

This approach ensures that only valid numeric entries remain, preventing bias in later statistical analysis.

***b. Data Type Validation***

All columns were explicitly converted to their proper data types:

* VALUE → float64 (numeric values)
* TIME\_PERIOD → datetime64[ns] (standardized date format)
* Metadata columns (e.g., geo, unit, na\_item) → string
* Invalid entries (e.g., text or special characters) were coerced to NaN and automatically removed.

***c. Outlier Identification***

Outlier detection was not systematically performed at this stage, as the main objective of Phase 2 was to ensure data consistency and structural integrity before analysis. A detailed investigation of extreme values will be conducted in Phase 3 using descriptive statistics and visual techniques such as boxplots and time-series plots.

***d. Feature Engineering***

No new features were created at this stage, as the focus was on data cleaning and preparation. Feature engineering will be addressed in the Phase 3, where derived indicators such as growth rates, normalized trade volumes (per capita or as GDP share), and relative transport intensities may be introduced to better capture the relationships specified in the research questions.

**3. Data Transformation and Harmonization**

***a. Standardizing Time Formats***

Eurostat datasets often represent periods in heterogeneous formats:

* Annual (YYYY)
* Quarterly (YYYY-Q)
* Monthly (YYYY-MM)
* Semestrial (YYYY-S)

A dedicated formatting script ([/src/format\_time\_periods.py](../src/format_time_periods.py)) was created to standardize these into a unified datetime structure.

The following logic was applied:

|  |  |  |
| --- | --- | --- |
| **Original Format** | **Example** | **Converted Date** |
| YYYY | 2023 | 2023-01-01 |
| YYYY-MM | 2023-05 | 2023-05-01 |
| YYYY-Qn | 2023-Q2 | 2023-04-01 |
| YYYY-Sn | 2023-S2 | 2023-07-01 |

This ensures full temporal consistency and allows easy alignment of monthly, quarterly, and annual series on a common timeline.

***b. Sorting and Structuring***

Each dataset was chronologically sorted by TIME\_PERIOD and saved in [/data/processed/formatted\_time\_periods/](../data/processed/formatted_time_periods).

This step guarantees consistent ordering for time series plotting and correlation analysis in the EDA phase.

***c. File Naming and Traceability***

Processed files retained the naming pattern of the original datasets with suffixes indicating transformation stages:

\_raw.csv → \_long.csv → \_formatted.csv

This traceability makes it possible to reproduce or audit every transformation step.

**4. Code Documentation and Reproducibility**

All scripts include detailed inline comments and are organized into logical steps corresponding to the assignment requirements:

[collecting\_data.py](../src/collecting_data.py) – automated data acquisition via Eurostat API CSV loading (in case of World Bank statistics it was done manually).

[transform\_to\_long\_format\_Estat.py](../src/transform_to_long_format_EStat.py) – conversion of Eurostat tables to long format.

[transform\_to\_long\_format\_WB.py](../src/transform_to_long_format_WB.py) – processing of World Bank datasets to long format.

[format\_time\_periods.py](../src/format_time_periods.py) – harmonization of time columns, zero-value filtering, and sorting.

The pipeline can be re-run end-to-end on any machine with Python ≥ 3.10 and the pandas library installed. A README.md file describes dependencies and execution instructions.

**5. Summary of Cleaning Outcomes**

After data cleaning and preparation:

* All 15 datasets are now standardized into a long, tidy, and machine-readable format.
* Zero and missing values were removed, leaving only valid numeric observations.
* Time columns are uniformly expressed as datetime, allowing cross-frequency alignment.
* The final cleaned datasets are stored in [/data/processed/formatted\_time\_periods/](../data/processed/formatted_time_periods), ready for Phase 3 (Exploratory Data Analysis).

This structured workflow establishes a solid foundation for subsequent correlation analysis, trend exploration, and visualization of Latvia’s economic and social indicators.

**Phase 3: Exploratory Data Analysis (EDA) & Visualization**

Since several Eurostat and World Bank indicators contain disaggregated categories (e.g., by gender, transport type, or age group), all observations were aggregated by time period to obtain total values per indicator. This ensures comparability across datasets and simplifies correlation analysis, as the research focuses on macroeconomic dynamics rather than intra-category differences.

**Rationale for Aggregation:**

Annual aggregation: The original datasets include monthly or quarterly values with different categories (gender, transport type, age group). For the strategic analysis of RQ1–RQ3, overall trends in the economy, migration, and transport flows are more relevant than intra-category differences. Aggregating by year reduces noise and enables consistent comparison across indicators with varying periodicity.

**Summing vs. averaging:**

Summed indicators: Transport volumes and economic measures (GDP, Exports) are summed to obtain total annual volumes.

Averaged indicators: Discrete metrics such as migration and unemployment are averaged to maintain representative annual values without losing data due to missing periods.

Interpolation for continuous indicators: Continuous economic and transport indicators require uninterrupted time series for trend analysis. Linear interpolation is applied to fill gaps between known values without artificially altering trends.

Handling missing values in discrete indicators: Migration, emigration, and unemployment are event-based and not always reported monthly. Filling these with interpolation could distort actual trends; therefore, missing values are left as is.

**Outcome:**

A single, annualized dataset with contextualized indicators (GDP, Population, Migration, Inflation, Transport) is produced, ready for analysis of RQ1–RQ3.

This approach reduces noise and duplicated information, simplifying visualization, correlation analysis, and multivariate exploration.

For each research question, exploratory analysis was conducted through univariate, bivariate, and correlation-based methods.  
Time series plots were used to identify long-term dynamics and structural breaks, while scatter plots highlighted direct relationships between key indicators.  
Correlation heatmaps provided a comprehensive view of interdependencies between variables, supporting the interpretation of trends and co-movements in Latvia’s economy.

This level of analysis is sufficient for identifying the direction and strength of associations relevant to the formulated research questions, without overcomplicating interpretation with excessive multivariate visualizations.

Multivariate plots were considered but omitted for clarity, as overlapping series with different scales reduced interpretability. Instead, relationships were effectively captured through pairwise scatter plots and correlation heatmaps.