# Lab 02: Game Dynamics Analysis with Python Classes

Course: Programming for Data Science
Professor: Dr. Carlos J. Costa

February 5, 2026

In this lab, you will take on the role of a data analyst studying game dynamics using the `GameTurns.xlsx` dataset, available at: https://github.com/masterfloss/FakeNewsData/raw/refs/heads/main/GameTurns.xlsx. The dataset contains information about game rounds, players' followers, credibility, and changes over time.

Your task is to design and implement Python classes that load, analyze, and visualize this data. You will practice object-oriented programming, data manipulation with pandas, and data visualization with matplotlib and seaborn.

## Task 1: File Data Reader

The first task is to create a Python class that reads the file from the given URL and loads its first worksheet into a pandas DataFrame. The class should be able to handle errors gracefully, such as when the file cannot be accessed or the URL is invalid. After implementing the class, you should test it by loading the dataset and displaying the first few rows to verify the import.

The expected output is a pandas DataFrame containing the data from the first worksheet of the file.

## Task 2: Linear Regression Analyzer

Next, you will create a Python class to perform Ordinary Least Squares (OLS) linear regression on a pandas DataFrame. The class should accept a DataFrame during initialization and provide a method where you can specify the dependent variable and a list of independent variables. This method should return the regression summary.

You should then apply this class to the dataset to investigate how variables such as `round_number`, `credibility_change`, and `followers_change` influence `current_followers`. The expected output is a `statsmodels` regression summary object.

## Task 3: Correlation Analyzer

The third task is to implement a Python class that calculates the correlation matrix for all numerical variables in the DataFrame. The class should also provide a method to visualize the correlation matrix as a heatmap, making it easier to interpret relationships between variables.

The expected results are a pandas DataFrame representing the correlation matrix and a heatmap plot displaying the correlation patterns.

## Task 4: Descriptive Statistics by Group

In this task, you will create a class that computes descriptive statistics for numerical columns, grouped by a specified column, such as `round_number`. The class should include methods to calculate the mean and standard deviation of the numerical columns, as well as a method to plot these statistics. The plot should display the mean as a line with a shaded area representing the standard deviation range.

Applying this class to the dataset will allow you to analyze how key metrics change over game rounds. The expected output is a DataFrame with grouped statistics and a plot illustrating the mean and standard deviation.

## Task 5: Refactoring and Integration

Finally, you should review all the classes you implemented in the previous tasks and refactor them to improve code structure, modularity, and reusability. Consider merging classes with overlapping functionality, creating helper methods for repeated operations, or introducing new classes where appropriate. The final result should be a coherent Python module or notebook where all classes work seamlessly together with well-organized methods.

## Submission Requirements

Students are expected to submit a file containing their implementation. This can be either a Jupyter Notebook (`.ipynb`) or a Python script (`.py`). In addition to the code, students should be prepared to explain their class design choices, how they implemented the analyses, and how the visualizations help interpret the dataset.