

# Financial Data Dashboard Project Syllabus

## Introduction

This project aims to create a comprehensive financial data dashboard integrating key metrics from Equities, Fixed Income, and Derivatives. The project focuses on refining skills in market knowledge, Excel proficiency, and Python (NumPy & Pandas). The purpose of this project is to simulate real-world financial analysis and demonstrate the ability to manipulate large datasets, perform complex financial calculations, and present the results in a user-friendly interface. By completing the project, you will refresh critical skills in data processing, financial modeling, and visualization, all relevant to the role of an Operations Analyst in Equity, Fixed Income, and Derivatives.

## 1. Python (NumPy & Pandas): Data Aggregation and Analysis

This section focuses on using Python to process raw financial data before importing it into Excel for further analysis.

We will collect historical data for equities, fixed income, and derivatives over a 5-year period (e.g., January 2019 - January 2024) and perform necessary calculations and cleaning operations.

- Task 1.1: Equities Data Cleaning & Analysis (5-year period)
  - Use Pandas to load and clean stock price data for 10 companies.
  - Functions to use: `pd.read_csv`, `dropna()`, `fillna()`, `apply()`
  - Calculate daily returns, annualized returns, and volatility.
  - Functions to use: `pct_change()`, `mean()`, `std()`
  - Deliverable: Cleaned CSV with calculated metrics ready for Excel.
- Task 1.2: Bond Data Processing (5-year period)

- Load bond data into Pandas and calculate yield, duration, and convexity.
  - Functions to use: ``pd.read_csv``, ``apply()``, ``lambda``
  - Simulate bond price changes based on interest rate fluctuations.
  - Deliverable: Bond data CSV with calculated metrics.
- 
- Task 1.3: Options Pricing
    - Implement Black-Scholes model to price European options.
    - Functions to use: ``exp()``, ``log()``, ``norm.cdf()``
    - Deliverable: CSV of option prices for various strike prices and expirations.
- 
- Task 1.4: Exporting Data to Excel
    - Use Pandas to export all cleaned and calculated datasets back into Excel.
    - Functions to use: ``pd.to_csv``, ``ExcelWriter()``
    - Deliverable: Exported CSV files and ready-to-use Excel inputs.

## **2. Excel Proficiency: Building the Financial Dashboard**

Once the data is processed using Python, we will use Excel to build a dashboard to analyze the equities, fixed income, and derivatives metrics.

The dashboard will allow for dynamic interaction and provide valuable insights into asset performance over time.

- Task 2.1: Equities Data Sheet & Metrics Calculation
  - Load the Python-generated CSVs into Excel.
  - Functions to use: ``VLOOKUP()``, ``IF()``, ``AVERAGEIF()``, ``STDEV()``, and ``SUMIF()``
  - Calculate total return, volatility, beta, and Sharpe ratio for each equity.
  - Deliverable: Automated Excel sheet calculating these metrics.

## - Task 2.2: Fixed Income Sheet with Bond Metrics

- Load bond data CSV from Python into Excel and calculate yield, duration, and convexity using formulas.

- Functions to use: `NPV()`, `RATE()`, `IRR()`

- Deliverable: Bond data sheet with real-time updates based on interest rate scenarios.

## - Task 2.3: Derivatives Sheet with Options Pricing

- Import Python-generated option prices and calculate payoffs for different strategies (e.g., covered calls).

- Functions to use: `IF()`, `MAX()`, `MIN()`

- Deliverable: Dynamic option pricing and payoff calculation sheet.

## - Task 2.4: Financial Dashboard Overview

- Combine equities, bonds, and derivatives sheets into a single, user-friendly dashboard.

- Use pivot tables and slicers to dynamically filter and display data.

- Incorporate correlation matrix between equities, bonds, and derivatives.

- Deliverable: A full financial dashboard summarizing asset performance and correlations over the past 5 years.