SEN605 · WEEK 1

Introduction to Systems Engineering and Financial Systems

Defining financial markets through a systems lens · Python ecosystem foundations

Instructor	Course	Semester	Delivery
Assoc. Prof. Onur POLAT	SEN605 — Elective	Fall / Spring	Face-to-Face

Week 1 — Agenda

What we will cover today

Course Overview

SEN605 — Financial Systems and Data Analytics

Course Objective

Teach advanced data analytics, machine learning, and time-series methods for financial systems

Equip students to formulate research questions from financial data

Develop predictive models using deep learning algorithms

Analyze and interpret system behavior in complex financial environments

Prepare a complete, publishable-quality research project

Learning Outcomes

Define financial system structure from a systems engineering perspective

Analyze financial time series: stationarity, volatility modeling

Build ML & deep learning forecasting models (LSTM, RF, XGBoost)

Perform risk measurement (VaR) and portfolio optimization

Design and present an academic-level research project

Prerequisites & Expectations

Basic programming knowledge (Python preferred, MATLAB/R acceptable)

Introductory statistics and probability (hypothesis testing, distributions)

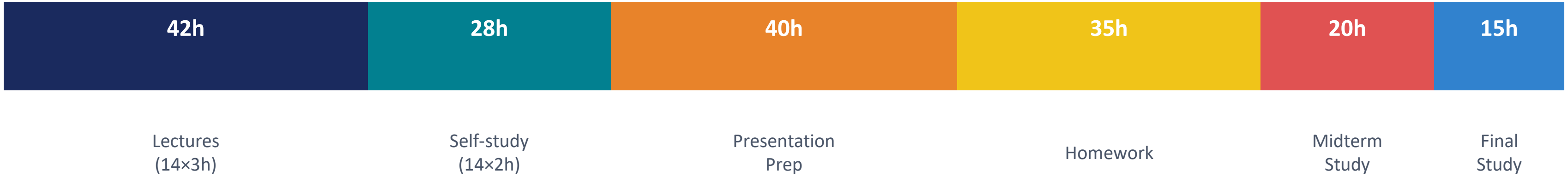
Willingness to engage with both theory and hands-on coding

Assessment & Workload

How your grade is calculated · 6 ECTS = 180 hours total

20%	30%	50%
Presentation (1 graded project)	Midterm Exam (Week 8)	Final Exam (End of semester)

Workload Distribution (Total: 180 hours)



Course Roadmap

14-week journey from fundamentals to research project

1	Intro to Systems Engineering & Financial Systems
2	Financial Data Structures & Python Ecosystem
3	Time Series Fundamentals & Stationarity
4	Volatility Modeling (GARCH Family)
5	Machine Learning for Finance I (Tree-Based)
6	Machine Learning for Finance II (Ensemble)
7	Deep Learning I: ANN, MLP, LSTM
8	**Midterm Exam**
9	Deep Learning II: Advanced Architectures
10	Risk Measurement: VaR & Expected Shortfall
11	Portfolio Optimization & Factor Models
12	Connectedness & Systemic Risk
13	Student Project Presentations
14	**Review & Final Exam Prep**

Section 1

Why Systems Engineering for Finance?

Connecting two disciplines that were made for each other

What is Systems Engineering?

A structured approach to designing and understanding complex systems

Definition: Systems Engineering is an interdisciplinary field that focuses on designing, integrating, and managing complex systems over their life cycles — ensuring all parts work together to achieve the intended purpose.

Holistic Thinking

Look at the whole system, not just individual parts. Interactions between components matter as much as the components themselves.

Model-Based Design

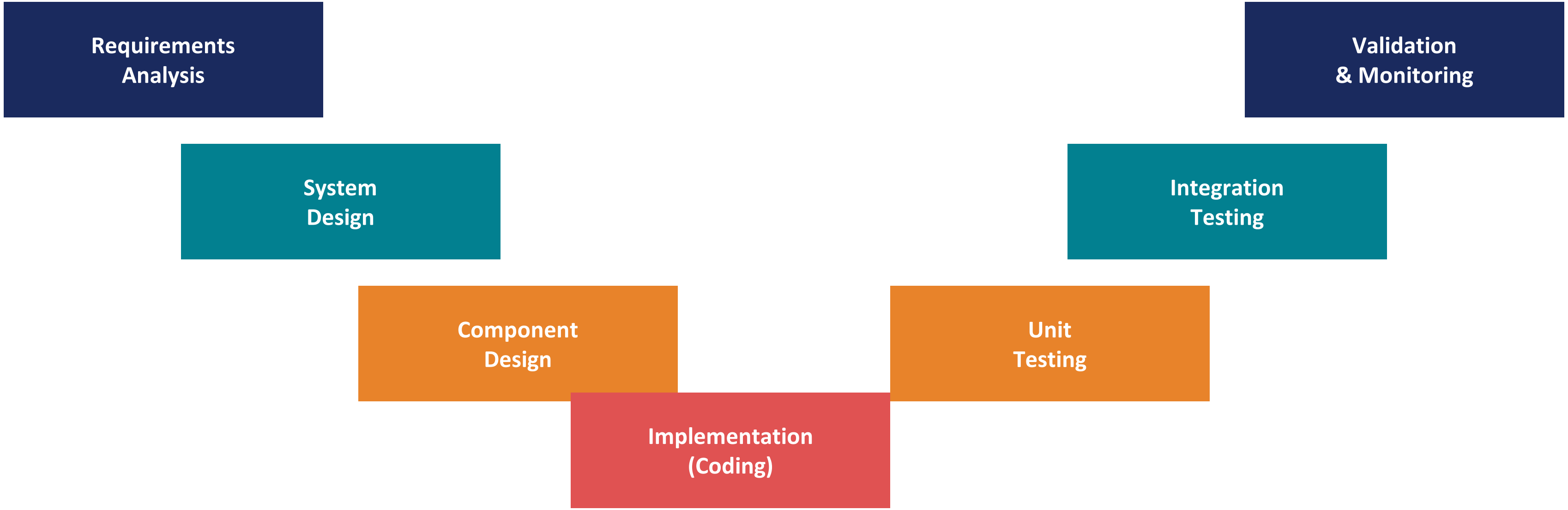
Use mathematical and computational models to represent, simulate, and test systems before deployment.

Lifecycle Perspective

Manage systems from concept through operation — including feedback and continuous improvement.

SE Lifecycle Models

How structured development applies to financial system design



Finance mapping: Requirements = Research question | Design = Model architecture | Implementation = Code | Testing = Backtest | Validation = Out-of-sample

Why SE for Finance Now?

Four forces making systems thinking indispensable

Algorithmic Trading

Over 70% of US equity volume is algorithmic. HFT systems require rigorous design, testing, and monitoring — classic SE territory.

Regulatory Complexity

Basel III/IV, MiFID II, DORA — financial regulation demands traceability, auditability, and lifecycle management of models.

Data Explosion

Alternative data (satellite, NLP, social) overwhelms ad-hoc analysis. Systematic pipelines require engineering discipline.

Interconnected Risk

2008, COVID, SVB — crises propagate through networks. Understanding systemic risk requires systems-level analysis.

Types of Systems

Simple vs. Complicated vs. Complex

Simple

e.g., A water pump

Few components
Linear cause-and-effect
Fully predictable
Expert can master it
Best practices apply

Complicated

e.g., A jet engine

Many components
Multiple right answers
Analyzable by experts
High expertise needed
Good practices apply

Complex

e.g., Financial Markets

Emergent behavior
Nonlinear dynamics
Adaptive agents
Perpetually novel
Emergent practices only

Financial markets are Complex — they resist prediction, exhibit emergence, and continuously evolve. This is why standard engineering tools must be extended.

Section 2

Financial Markets as Complex Systems

Four defining properties of complex adaptive systems

Property 1: Nonlinearity

Small causes can produce disproportionately large effects

Linear System

Input x 2 → Output x 2

Predictable and proportional

Superposition principle holds

Classic physics & engineering

Example: A simple spring — double the force, double the extension (Hooke's Law).

Nonlinear System (Finance)

Input x 2 → Output x 50

Flash crashes from trivial triggers

Threshold and tipping-point effects

Black swan events are predictable in frequency, not timing

Example: 2010 Flash Crash — an algo error triggered a 1,000-point DJIA drop in minutes.

Implication for modeling: Linear regression and correlation may fail catastrophically during crises. Non-linear methods (tree-based ML, neural networks, quantile regression) are essential complements.

Property 2: Feedback Loops

Prices influence behavior — behavior influences prices

Positive Feedback (Amplifying)

Price rises → more buyers enter → price rises further → bubble

Amplifies deviations from equilibrium

Drives bubbles and crashes

Markets can stay irrational longer than you can stay solvent

Examples: Bitcoin 2017 bull run · Dot-com bubble · Housing crisis 2006–2008

Negative Feedback (Stabilizing)

Price rises → fewer buyers, more sellers → price stabilizes

Restores equilibrium

Classical economics relies heavily on negative feedback

Arbitrage and mean-reversion strategies exploit this

Examples: Arbitrage · Mean-reversion strategies · Central bank interventions

Property 3: Emergence

The whole is more than the sum of its parts

> *"No individual trader intends to create a market trend — yet trends emerge from millions of individual decisions."*

Market Trends

No single agent decides 'the market will trend up today'. Persistent trends emerge from aggregate pressure.

Herding Behavior

Rational individual decisions to follow consensus create irrational collective behavior — bank runs, panic selling.

Price Discovery

The 'correct' price emerges from millions of bids, asks, and trades in continuous auctions.

Systemic Risk

Individual banks manage risk carefully, yet the system itself becomes fragile. 2008: each bank was 'fine' until all were not.

Property 4: Adaptivity

Agents learn, evolve, and change the rules of the game

Why Adaptivity Matters

Markets adapt to strategies — any profitable pattern eventually disappears as traders exploit it

Regulations change behavior, which demands new regulations

Technology (HFT, algorithmic trading) reshapes market microstructure

Investor psychology evolves: post-crisis behavior differs from pre-crisis

ML models trained on past data must retrain continuously — concept drift

1	**Observe**	Agent receives market signal
2	**Learn**	Update internal model / belief
3	**Act**	Execute trade or strategy
4	**Impact**	Price changes, market moves
5	**Repeat**	Cycle continues — system evolves

Network Structure in Finance

Financial systems as interconnected graphs

Key Concepts

Nodes = banks, funds, firms, countries

Edges = lending, trade, ownership, correlation

Degree centrality: who is most connected?

Betweenness: who bridges clusters?

Clustering coefficient: are neighbors also connected?

Why Networks Matter

Contagion: failure of one node cascades through edges

Too-interconnected-to-fail (not just too-big)

Network topology predicts systemic risk better than individual metrics

Lehman Brothers: a hub failure, not just a node failure

Central banks now monitor financial networks directly

Course Application

Week 12: Connectedness analysis using TVP-VAR and DY (Diebold-Yilmaz) spillover frameworks

Construct and visualize dynamic financial networks from return/volatility data

Measure time-varying systemic risk using network-derived indicators

Phase Transitions & Regime Changes

Markets shift abruptly between calm and crisis states

Calm Regime

Low volatility, mean-reverting returns, normal distribution approximation acceptable, standard correlations hold.

Transition Zone

Volatility picks up, correlations begin shifting, early-warning signals (LPPLS, entropy) may fire.

Crisis Regime

Extreme volatility, fat tails dominate, correlations spike toward 1.0 ("everything sells off together"), non-normal behavior.

Modeling implication: A single model trained on all data performs poorly. Regime-switching models (Markov, threshold VAR) or separate models per regime often outperform. We will encounter this in volatility modeling (Week 4) and risk measurement (Week 10).

Case Study: Turkish Financial System

Complexity in action — crises and structural shifts

2001 Banking Crisis	Overnight rates spiked to 7,500%. 20+ banks failed. Led to BRSA (BDDK) creation and major banking reform.
2018 Currency Crisis	TRY lost ~40% vs USD. Capital outflows, inflation surge. Demonstrated FX-equity-bond interconnectedness.
2021-23 Unorthodox Policy	Rate cuts amid high inflation. Extreme TRY depreciation. Natural experiment in policy-market feedback loops.
2023+ Policy Normalization	Return to orthodox monetary policy. Gradual disinflation. Foreign investor re-entry into Borsa Istanbul.

Each episode illustrates nonlinearity, feedback, emergence, and adaptivity — the four properties we just studied.

Section 3

Market Participants & Instruments

Who trades, what they trade, and why it matters for modeling

Market Participants

Every agent has different objectives, time horizons, and information

Institutional Investors

Pension funds, mutual funds, insurance companies
Motive: Long-term return, risk-adjusted performance
Info: Fundamental + macro research

High-Freq. Traders (HFT)

Citadel, Virtu, Jane Street
Motive: Microsecond profit from market microstructure
Info: Order flow, Level-2 data

Hedge Funds

Two Sigma, Renaissance, Bridgewater
Motive: Alpha generation, absolute return
Info: Quant models, alternative data

Central Banks

TCMB, Fed, ECB
Motive: Price stability, FX intervention
Info: Macro policy signals

Retail Investors

Individual traders, app-based trading
Motive: Wealth accumulation, speculation
Info: News, social media, TA

Corporations

Multinationals hedging FX/commodity risk
Motive: Risk management, not profit
Info: Internal exposure reports

Turkish Capital Markets: Borsa Istanbul

Structure, indices, and key characteristics

Market Structure

Borsa Istanbul (BIST): sole exchange since 2013 merger

Equity, debt, derivatives, precious metals markets

Central Securities Depository: MKK (Merkezi Kayit Kurulusu)

Settlement: T+2 for equities

Trading hours: 10:00–18:00 (continuous auction)

Key Indices

BIST-100 (XU100): Broad market benchmark

BIST-30 (XU030): Large-cap, most liquid

BIST-Bank (XBANK): Banking sector index

BIST-Industrial (XUSIN): Manufacturing sector

BIST-Sustainability (XUSRD): ESG-screened index

Key Characteristics for Modeling

High retail participation (~70% of trading volume) — sentiment effects are strong

Strong FX sensitivity: USD/TRY moves and BIST-100 are inversely correlated in crisis periods

Concentrated banking sector: XBANK often drives XU100

Short-selling restrictions and circuit breakers affect price dynamics and model assumptions

Financial Instruments

What gets traded — and what data each produces

Instrument	Example	Key Risk	Data Generated
Equities (Stocks)	AKBNK, AAPL	Market risk	OHLCV, dividends, splits
Fixed Income (Bonds)	Treasury bonds, corporate bonds	Interest rate, credit risk	Yield, duration, spread
Foreign Exchange (FX)	USD/TRY, EUR/USD	Currency risk	Spot rates, swaps, vol
Derivatives	Options, futures, swaps	Leverage, counterparty	Strike, expiry, Greeks
Commodities	Gold, oil, wheat	Supply/demand, geopolitical	Spot, futures, inventory
Crypto Assets	BTC/USD, ETH/USD	Extreme volatility, regulatory	Tick, on-chain metrics

23

Market Microstructure

How prices are actually formed — the mechanics behind the data

Key Microstructure Concepts

Bid-Ask Spread: difference between best buy and sell price — measures liquidity

Order Book: queued buy/sell orders at different prices — real-time supply/demand

Market Impact: large orders move prices — a challenge for institutional traders

Price Discovery: the process by which information becomes reflected in prices

Execution Venues: exchanges, dark pools, OTC — affect data availability

Tick Size: minimum price movement — varies by exchange and instrument

Why This Matters for ML Models

Bar construction from tick data to OHLCV requires microstructure decisions

Bid-ask bounce creates false serial correlation in high-frequency data

Execution costs (slippage) must be modeled for realistic backtests

Order book features (imbalance, depth) are powerful predictive signals

Regime changes in microstructure invalidate models trained on old data

Information Asymmetry & Market Efficiency

The Efficient Market Hypothesis and its limits

Weak Form

Prices reflect all past trading data. Technical analysis cannot earn excess returns. Random walk hypothesis.

Tests: Serial correlation tests, runs tests, variance ratio

Semi-Strong Form

Prices reflect all publicly available info. Fundamental analysis cannot beat market. Event studies test this.

Tests: Event studies, calendar anomalies, factor models

Strong Form

Prices reflect all info, including insider knowledge. No one can earn excess returns. Rarely holds in practice.

Tests: Insider trading studies, hedge fund performance

Practical stance: Markets are efficient enough that naive strategies fail, but inefficient enough that sophisticated ML/DL models can find transient alpha — especially in emerging markets like Borsa Istanbul where information flows are slower.

Regulatory Framework

Key regulators and their impact on data and modeling

SPK (CMB)

Capital Markets Board of Turkey. Securities regulation, disclosure requirements, short-selling rules, market abuse enforcement.

TCMB

Central Bank of Turkey. Monetary policy, interest rate decisions, FX interventions, reserve requirements. Major market driver.

BDDK (BRSA)

Banking Regulation and Supervision Agency. Capital adequacy, stress testing, systemic risk monitoring for Turkish banks.

SEC / Fed (US)

Comparable US regulators. Many datasets (EDGAR, FRED) and regulations (Reg NMS, Dodd-Frank) affect global modeling.

Section 4

Financial Data Landscape

Types, sources, and quality considerations

Types of Financial Data

From raw ticks to alternative datasets

Tick Data	*Every trade / quote*	Price: 142.35, Volume: 500, Time: 09:31:45.231 + Highest resolution, full microstructure | - Massive storage, complex cleaning
Bar Data (OHLCV)	*1min, 5min, Daily*	Open: 142.0, High: 143.5, Low: 141.8, Close: 143.2, Vol: 1.2M + Compact, industry standard, widely available | - Sampling artifacts; ignores intrabar moves
Fundamental Data	*Quarterly / Annual*	P/E: 12.3, EPS: 4.5, Revenue: 2.1B TRY, Debt/Equity: 0.4 + Long-term value signals | - Slow-moving; reporting lags (look-ahead risk)
Alternative Data	*Varies*	Satellite images of parking lots, credit card transactions, sentiment + Information edge; not yet priced in | - Expensive, noisy, requires heavy preprocessing

Data Quality & Common Pitfalls

Garbage in, garbage out — data cleaning is 80% of the work

Missing Data

Holidays, halts, delisted stocks create gaps. Forward-fill? Interpolate? Drop? Each choice biases results differently.

Survivorship Bias

Datasets that include only current stocks (survivors) overstate historical returns. Failed companies disappear from history.

Look-Ahead Bias

Using information that wasn't available at the time of the trading decision. Common with fundamental data (reporting lags).

Corporate Actions

Stock splits, dividends, mergers change price levels. Adjusted prices (auto_adjust=True) are essential for return calculations.

Time Zone & Calendar

BIST trades in Istanbul time, NYSE in New York. Aligning multi-market data requires timezone-aware operations.

Currency Effects

Comparing BIST (TRY) with S&P 500 (USD) without currency adjustment leads to misleading conclusions.

Data Vendors & Free Sources

Where to find financial data for research

Free Sources (Course Default)

Yahoo Finance (via yfinance): equities, FX, crypto, indices

FRED (Federal Reserve): macro indicators, interest rates, inflation

TCMB EVDS: Turkish central bank electronic data system

KAP (Public Disclosure Platform): Borsa Istanbul disclosures

World Bank, IMF: global macro and development data

Commercial Sources (FYI)

Bloomberg Terminal: gold standard for financial data

Refinitiv (LSEG): formerly Thomson Reuters Datastream

Quandl (Nasdaq): alternative and financial data APIs

CryptoCompare / Kaiko: crypto market data

S&P Capital IQ: fundamentals and screening

Academic & Research Sources

WRDS (Wharton Research Data Services): CRSP, Compustat, TAQ — many universities have access

Kenneth French Data Library: Fama-French factor returns (free, essential for factor research)

ADS (Aruoba-Diebold-Scotti): Philadelphia Fed business conditions index

EPU (Economic Policy Uncertainty): Baker-Bloom-Davis index (free, multiple countries)

Survivorship & Look-Ahead Bias

Two critical traps that invalidate backtests

Survivorship Bias

What it is: Only studying companies/assets that exist today, ignoring those that failed, delisted, or merged.

Average return of survivors is always higher than the true market return

Hedge fund databases: only successful funds report → overstated average performance

BIST example: companies delisted after bankruptcy disappear from XU100 history

Fix: Use point-in-time constituent lists. Include delisted companies with their full history.

Look-Ahead Bias

What it is: Using data in a model that would not have been available at the time of the decision.

Quarterly earnings announced 30-60 days after quarter end — using them on quarter-end date is look-ahead

Using future data to normalize (StandardScaler on full dataset, then split train/test)

Feature engineering using the entire time series (e.g., rolling z-score with future values)

Fix: Use expanding windows. Always split BEFORE feature engineering. Respect publication dates.

Section 5

Python Ecosystem for Finance

The tools we'll use throughout this course

Python Ecosystem Overview

Core libraries for financial data science

Data Layer Acquire and ingest raw financial data	yfinance	pandas_datareader	FRED API	CCXT (crypto)	KAP scraper
Processing Layer Clean, transform, engineer features	pandas	numpy	scipy	polars	
Modelling Layer Classical ML and statistical models	scikit-learn	statsmodels	arch	xgboost	
Deep Learning Layer Neural networks, LSTM, FinBERT	PyTorch	TensorFlow	Keras	transformers	
Visualization Layer Charts, dashboards, interactive plots	matplotlib	seaborn	plotly	mplfinance	

Key Library: pandas

The backbone of financial data manipulation in Python

What pandas provides

DataFrame and Series — labeled, time-indexed tabular data

Date parsing, resampling, and timezone-aware operations

Rolling, expanding, and ewm windows for time series

GroupBy, merge, pivot — all the SQL operations you need

Finance-specific operations

resample('W') — weekly OHLCV bars from daily data

pct_change() / diff() — return series construction

rolling(21).std() — rolling volatility windows

merge_asof() — align datasets with different frequencies

Code Snippet

```
import pandas as pd
import yfinance as yf

# Download BIST-100 data
df = yf.download("XU100.IS",
    start="2020-01-01",
    end="2024-12-31")

# Compute daily log returns
df['log_ret'] = df['Close'].apply(
    lambda x: np.log(x)).diff()

# 21-day rolling volatility
df['vol21'] = (df['log_ret']
    .rolling(21).std()
    * np.sqrt(252))

# Weekly resampled OHLCV
weekly = df.resample('W').agg({
    'Open':'first','High':'max',
    'Low':'min','Close':'last',
```

Key Libraries: numpy & matplotlib

Numerical computation and financial visualization

numpy — numerical foundation

Fast array operations: returns matrix, covariance matrices

log(), exp(), sqrt() — essential for financial transformations

random — Monte Carlo simulations for VaR, option pricing

linalg — matrix inversion for portfolio optimization

matplotlib / seaborn — visualization

Price and return series plots with dual-axis

Heatmaps: correlation matrices of multi-asset portfolios

Histograms with KDE overlays: return distributions

mplfinance: candlestick and OHLC financial charts

 # Code Snippet

```
import numpy as np
import matplotlib.pyplot as plt

# Log returns (numpy)
prices = df['Close'].values
log_ret = np.log(prices[1:] /
                 prices[:-1])

# Annualized volatility
ann_vol = log_ret.std() * np.sqrt(252)
print(f"Vol: {ann_vol:.2%}")

# Plot return distribution
fig, ax = plt.subplots(figsize=(8,4))
ax.hist(log_ret, bins=80,
        color='#1A2A5E',
        alpha=0.75,
        edgecolor='white')
ax.set_title('Return Distribution')
ax.set_xlabel('Log Return')
plt.tight_layout()
```

Key Libraries: ML & Deep Learning

What each library handles in this course

scikit-learn

Weeks 5–6

Linear/Logistic regression
Random Forest, XGBoost wrappers
TimeSeriesSplit for CV
StandardScaler, preprocessing
Metrics: RMSE, AUC-ROC, Sharpe

statsmodels

Weeks 3–4

ARIMA / SARIMA estimation
ADF, KPSS stationarity tests
Ljung-Box diagnostic tests
OLS with full statistical output
VAR models for multivariate TS

arch

Week 4

GARCH / EGARCH / GJR-GARCH
ARCH LM test
DCC multivariate volatility
Conditional covariance estimation

PyTorch

Weeks 7, 9

MLP / ANN from scratch
LSTM, GRU, RNN modules
Custom training loops
GPU acceleration
Autograd — automatic differentiation

36

Jupyter & Reproducible Research

Best practices for computational finance notebooks

Notebook Best Practices

One notebook per analysis/experiment — don't mix unrelated work

Top cell: imports + configuration (random seeds, display options)

Narrative flow: explain WHY, not just WHAT the code does

Use Markdown cells for section headers and interpretation

Restart kernel and run all before submission — order matters!

Reproducibility Checklist

Set random seeds: np.random.seed(42), torch.manual_seed(42)

Pin library versions: pip freeze > requirements.txt

Save raw data separately — never modify source files in place

Log all hyperparameters and model configurations

Use relative paths: Path('data/') not '/Users/onur/data/'

Project Structure (Recommended)

project/ → data/ (raw + processed) | notebooks/ (exploration + final) | src/ (reusable modules)

project/ → results/ (figures + tables) | README.md (what, how, why) | requirements.txt

Version control with Git: commit early, commit often, write meaningful messages

Section 6

Hands-On: First Data Pull

Downloading, exploring, and plotting real financial data

Environment Setup

Getting your Python environment ready for financial work

1

Install Anaconda or Miniconda

```
Download from: https://www.anaconda.com/download
Recommended: Python 3.10+ distribution with conda package manager
```

2

Create a virtual environment

```
conda create -n sen605 python=3.11
Isolates project dependencies from your system Python
```

3

Activate and install core packages

```
conda activate sen605
pip install yfinance pandas numpy matplotlib seaborn
pip install scikit-learn statsmodels arch xgboost
pip install torch torchvision transformers
```

4

Launch Jupyter Lab

```
pip install jupyterlab
jupyter lab
Access notebooks at http://localhost:8888
```

Downloading Financial Data with yfinance

Accessing BIST-100, S&P 500, FX, and crypto data in one line

What yfinance provides

Historical OHLCV data from Yahoo Finance API (free)

Supports equities, indices, FX, crypto, ETFs

Automatic currency adjustment (auto_adjust=True)

Intraday data at 1m/5m/15m/60m intervals (last 60 days)

Key tickers for this course

XU100.IS — BIST-100 Index (Turkish equities)

XU030.IS — BIST-30 (large-cap Turkish stocks)

USDTRY=X — USD/Turkish Lira exchange rate

^GSPC — S&P 500 (U.S. equities)

BTC-USD — Bitcoin vs. USD (crypto)

GC=F — Gold futures (commodity)

 # Week 1 — Lab Code

```python
import yfinance as yf
import pandas as pd

# Single ticker download
bist = yf.download(
    "XU100.IS",
    start="2020-01-01",
    end="2024-12-31",
    auto_adjust=True
)

# Multiple tickers at once
tickers = ["XU100.IS","USDTRY=X",
           "^GSPC","BTC-USD"]
data = yf.download(
    tickers,
    start="2020-01-01",
    end="2024-12-31",
    auto_adjust=True
)
```

Exploring the Data

Basic inspection — always do this before any modeling

Data Inspection Checklist

.shape — how many rows x columns?

.dtypes — are dates parsed correctly?

.isnull().sum() — any missing data?

.describe() — min, max, mean, std, quartiles

.head() / .tail() — sanity-check first/last rows

Plot raw price — obvious outliers? Gaps?

Check corporate actions — stock splits, dividends

Verify trading hours — remove non-trading zeros

 # Exploration code

```
# Basic info
print(bist.shape)
print(bist.dtypes)

# Missing data check
print(bist.isnull().sum())

# Descriptive stats
print(bist.describe().round(2))

# Check date range
print("Start:", bist.index[0])
print("End:  ", bist.index[-1])
print("Days:", len(bist))

# Sanity: plot closing price
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10,4))
bist['Close'].plot(ax=ax,
    color='#1A2A5E', lw=1.2)
ax.set_title('BIST-100 Close')
```

Data Cleaning Pipeline

A systematic approach to preparing financial data

1 Load & Inspect

Read raw data, check types, examine first/last rows

2 Handle Missing Values

Identify pattern (random vs structural), choose strategy (ffill, drop, interpolate)

3 Remove Outliers

Flag returns > 3-4 std devs, check if genuine or data error

4 Adjust for Corporate Actions

Use adjusted prices, handle splits and dividends

5 Align Time Series

Common date index for multi-asset analysis, handle timezone differences

6 Feature Engineering

Compute returns, volatility, rolling stats on cleaned data

Rule of thumb: Data cleaning and preparation typically takes 60-80% of total analysis time. Budget accordingly.

Handling Missing Data

Strategies and trade-offs for financial time series

Forward Fill (ffill)

Carry last known value forward. Best for: prices on non-trading days.

+ Simple, preserves level | - Stale during long gaps

Drop (dropna)

Remove rows with any missing values. Best for: sporadic gaps in returns.

+ No fabricated data | - Reduces sample size, misaligns assets

Interpolation

Linear or spline between known values. Best for: smooth macro series (GDP, inflation).

+ Smooth series | - Assumes linearity between points

Model-Based

EM algorithm, KNN, or regression imputation. Best for: multivariate panels with patterns.

+ Uses cross-asset info | - Complex, risk of overfitting

Section 7

Returns & Volatility: A First Look

From prices to the language of finance

Computing Returns

Simple returns vs. log-returns — when to use which

Simple Return

$$r_t = (P_t - P_{t-1}) / P_{t-1}$$

```
df['simple_ret'] = df['Close'].pct_change()
```

Advantages

Intuitive: 10% means +10%

Portfolio aggregation is simple (weighted sum)

Used in practice for reporting

Limitations

Not time-additive: (1+r1)(1+r2) != 1+r1+r2

Bounded at -100%, unbounded upward

Harder to use in statistical models

Log Return

$$r_t = \ln(P_t / P_{t-1})$$

```
df['log_ret'] = np.log(df['Close']/df['Close'].shift(1))
```

Advantages

Time-additive: r1 + r2 + r3 = total

Symmetric: +10% and -10% equal magnitude

Closer to normally distributed

Limitations

Not directly portfolio-aggregatable

Slightly harder to interpret

Approximation breaks for large returns

Return Distributions: Normal vs. Reality

Why the bell curve is a dangerous simplification

Normal Distribution Assumption

Mean and variance fully describe the distribution

Symmetric: upside = downside probability

Events beyond 3 sigma are extremely rare (0.3%)

Basis for many financial models (Black-Scholes, MPT, VaR)

Computationally convenient: closed-form solutions

Empirical Reality

Fat tails: extreme events are 10-100x more likely than normal predicts

Negative skewness: crashes are sharper than rallies

Excess kurtosis: peaked center + heavy tails

Time-varying parameters: mean and variance shift over time

BIST-100 kurtosis: typically 5-8 (normal = 3)

Key Statistics to Check

Skewness: measures asymmetry (0 = symmetric, < 0 = left tail heavier) | Kurtosis: measures tail thickness (3 = normal, > 3 = fat tails) | Jarque-Bera test: formal normality test ($p < 0.05$ → reject normality) | Q-Q plot: visual comparison against normal quantiles

Stylized Facts of Financial Returns

What every financial time series has in common

Fat Tails (Leptokurtosis)

Extreme returns occur far more frequently than a normal distribution would predict. Black swan events are not rare — just unpredictable in timing.

Volatility Clustering

Large returns (of either sign) tend to cluster together. Calm follows calm; turbulent follows turbulent. Motivates GARCH models.

Negative Skewness

Markets tend to fall faster than they rise. Equity returns are slightly negatively skewed — crashes are sharper than rallies.

Serial Uncorrelation

Return autocorrelation is near zero at most lags. But squared returns (volatility) show strong autocorrelation.

Leverage Effect

Volatility increases more after a price decrease than after an equivalent increase. Captured by EGARCH and GJR-GARCH.

Long Memory in Volatility

Volatility shocks decay very slowly — evidence of fractional integration. Relevant for long-horizon risk forecasting.

Correlation & Dependence Structures

Measuring co-movement between financial assets

Pearson Correlation

Linear relationship between returns. Range [-1, +1]. Most common, but assumes normality and linearity.

Use: Daily/weekly return correlations, portfolio construction

Spearman Rank Correlation

Monotonic relationship (not necessarily linear). Robust to outliers. Based on rank-order of observations.

Use: Non-normal data, ordinal variables, robustness checks

Tail Dependence

Do assets crash together? Standard correlation may miss this. Copulas measure joint extreme behavior.

Use: Systemic risk, crisis modeling, VaR estimation

Key insight: Correlations are not constant — they increase during crises ("diversification fails when you need it most"). Time-varying and regime-dependent correlation models are covered in Weeks 4 and 12.

Introduction to Risk Measures

A preview of what we'll formalize in Week 10

Volatility (Sigma)

Standard deviation of returns. The simplest risk measure. Annualized: sigma x sqrt(252). Limitation: treats upside and downside equally.

Value-at-Risk (VaR)

Maximum expected loss at a given confidence level (e.g., 95% VaR). "We are 95% confident we won't lose more than X." Does not tell you how bad it gets beyond X.

Expected Shortfall (ES)

Average loss beyond VaR threshold. Also called CVaR or Tail VaR. "When things go wrong, how wrong do they get on average?" More informative than VaR for fat-tailed distributions.

Maximum Drawdown

Largest peak-to-trough decline in portfolio value. Measures worst-case historical experience. Intuitive for investors: "How much could I lose?"

Week 1 Summary

Key takeaways from today's session

Systems Thinking

Financial markets are complex adaptive systems

Nonlinearity, feedback, emergence, and adaptivity are core properties

Network structure and regime changes matter

Market Structure

Multiple agent types with conflicting objectives

Microstructure determines data quality

Turkish markets: high retail, FX-sensitive, bank-dominated

Data Landscape

Tick → Bar → Fundamental → Alternative data pyramid

Data quality and cleaning are critical first steps

Survivorship and look-ahead bias can invalidate results

Python Toolkit

pandas + numpy: data manipulation foundation

scikit-learn + statsmodels: modeling layer

PyTorch + transformers: deep learning layer

Week 1 Assignment

Notebook Assignment (ungraded — for practice)

1. Download BIST-100, S&P 500, BTC/USD, and USD/TRY for the last 5 years
2. Compute daily log-returns and create a joint returns DataFrame
3. Compute annualized volatility (std x sqrt(252)) for each asset
4. Compute the pairwise correlation matrix and plot it as a heatmap
5. Which two assets are most and least correlated? Write 2-3 sentences interpreting why.

Next Week

Week 2: Financial Data Structures & Python Ecosystem

Deep dive into OHLCV bars

Feature engineering pipeline

Data cleaning best practices

Build your first factor model

"In God we trust; all others bring data." — W. Edwards Deming