Objective & Data Domains

The goal of this assignment is to understand the structure of time series in two different domains and to reproduce their typical patterns with the *Time Series Random Configurator* app.

For each domain, we describe the main characteristics of the original series, configure the simulator accordingly, compare the simulated series with the original ones, and discuss the strengths and weaknesses of the tool.

Domain descriptions & expected structure

1. Finance domain — original figure (Jan–May 2020, S&P 500 during COVID)

Horizon. About **140 trading days** (from early January to mid-May 2020).
Structural phases.

- Pre-crash: a slow upward drift with low volatility and small day-to-day oscillations.
- Crash: a prolonged drawdown (not a sharp V) of roughly −34% from the local peak, with the trough around 60–65% of the window (≈ 23 March). Volatility spikes (heteroskedasticity) and returns display negative skew.
- Post-crash: a gradual, incomplete recovery; the series does not reclaim the initial highs by the end of the window. Volatility remains elevated relative to the pre-crash period.

Time-series properties. Non-stationary levels (price), approximately stationary returns with volatility clustering; no deterministic seasonality at the daily horizon.

2. Hotel domain — original figure (weekly occupancy, “2015” line)

Horizon. 52 weeks (weekly sampling).
Seasonal pattern. Pronounced annual seasonality with a summer hump.

- Peak: weeks ~31–33 at ≈76–78%.
- Post-summer drop: weeks ~36–38 down to ≈66% (fast decline).
- Autumn shoulder: weeks ~43–46 back up to ≈69–71%.
- Year-end decline: weeks ~50–52 down to ≈42–45%.
- The curve is smooth, visually consistent with a short moving average (≈4-week) smoothing.

Time-series properties. Strong deterministic seasonality (annual), mild downward trend into Q4, low high-frequency noise; the mean level is around 60% with an amplitude of ≈±20 percentage points.

Simulation set-up and parameters in the app

1. Finance domain simulation

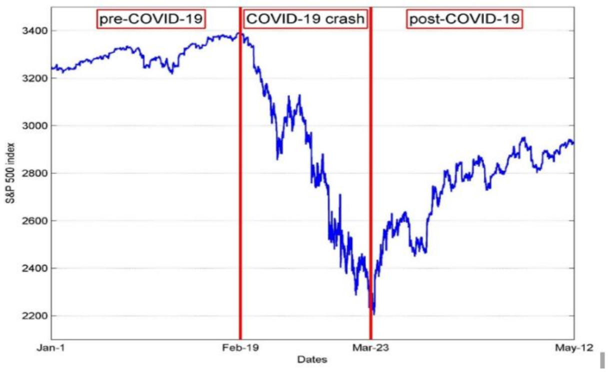


Figure 1 - (JRFM / MDPI): “Original figure of the market regimes” panel explicitly showing pre-COVID → crash → recovery; great if you want to mention *regime change* visually.

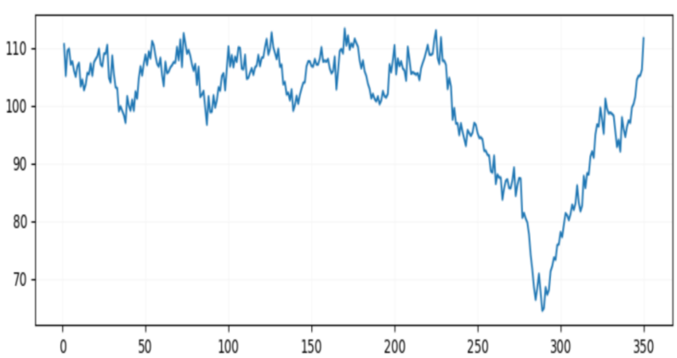


Figure 2 - Reproduction of time series random configurator of Jan–May 2020, S&P 500 during COVID.

Configuration used (Time Series Random Configurator):

- **Main**: Number of values = 350, Number of time series = 10, Mean = 105.0, Standard deviation = 0.0, Standard deviation of means = 0.0.
- **Trend**: Trend type = Increasing, Trend intensity = 2.7, Trend randomness = Yes, Trend randomness standard deviation = 2.0.
- **Peaks & Valleys**: Option = Valley, Magnitude = 34% (approximate size of the COVID drawdown), Probability of occurrence = 14% (≈ one crash in 350 points), Recovery = Yes, Recovery duration = 120-time units, Peak position in recovery = Middle.
- **Periodicity**: First periodicity = Sine, Amplitude = 3.0, Period = 52; Second periodicity = Cosine, Amplitude = 3.2, Period = 26.

2. Hotel domain simulation

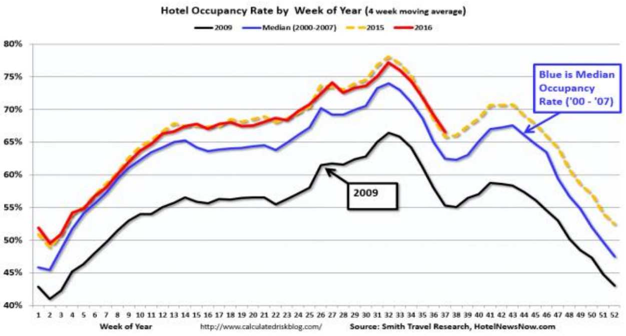


Figure 3 (Managerial Econ / STR): Original figure of the weekly/annual seasonal pattern in hotel occupancy, nice for explaining weekly profiles and the late-summer drop.

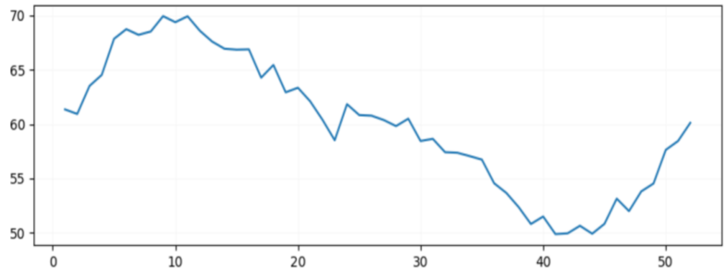


Figure 4: Hotel occupancy generated with Time Series Random Configurator.

Configuration used (Time Series Random Configurator):

- **Main:** Number of values = 52, Number of time series = 10, Mean = 60.0, Standard deviation = 0.0, Standard deviation of means = 0.0.
- **Trend:** Trend type = Increasing, Trend intensity = 1.9, Trend randomness = Yes, Trend randomness standard deviation = 2.0.
- **Peaks & Valleys:** Option = Both peaks and valleys, Magnitude = 2.8%, Probability of occurrence = 20%, Recovery = Yes, Recovery duration= 2, Recovery type = Both.
- **Periodicity:** First periodicity = Sine, Amplitude = 8.0, Period = 52.0; Second periodicity = Sine, Amplitude = 3.0, Period = 26.0.

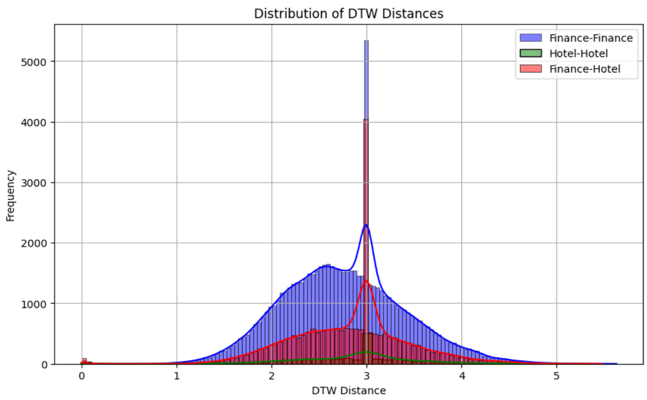
These settings enforce one long valley roughly in the middle of the series, combined with a mild upward trend and small oscillations before the crash. The randomness in the trend and in the valley recovers some of the irregularity observed in the empirical data.

Distances between time series

To quantify the similarity between series, we compute Euclidean distances between the simulated time series.
For each domain we generate 10 simulated series, obtaining two matrices of size $10 \times T$: one for finance and one for hotels. We then compute:

- Within-domain distances: all pairwise distances between series from the same domain (finance–finance, hotel–hotel).

- Between-domain distances: distances between each finance series and each hotel series. Because the two domains have different lengths, we use only the first $T^* = \min(T_{\text{finance}}, T_{\text{hotel}})$ time points when comparing across domains.



The results can be summarised by the **average distance** in each group:

- average distance within finance: **[blue in Figure]**
- average distance within hotels: **[green in Figure]**
- average distance between domains: **[red in Figure]**

Crucially, the **cross-domain distances** are consistently larger than the within-domain distances. This supports the idea that the simulator, despite its simplicity, generates two clearly distinct families of time series: one "crash-dominated" (finance) and one "seasonal" (hotels).

Conclusions: quality of the simulation and assessment of the tool

The experiment shows that the Time Series Random Configurator can successfully reproduce the **high-level behaviour** of time series from very different domains, although it cannot replicate all details of the real data.

How well were the series simulated?

- **Finance:** The simulated S&P 500 reproduces pre-crash growth, a deep prolonged crash, and partial recovery; crisis is enforced via valley settings, but volatility clustering/heavy tails/asymmetric jumps are only approximated, so the path is smoother, and trough timing depends ad-hoc on probability/duration.

- **Hotel:** The simulator captures strong annual seasonality with realistic levels—a summer peak and winter low plus small irregularities—but lacks phase control, so the peak misaligns with weeks 31–33 and the series is less smooth than the 4-week-averaged empirical curve. The **distance analysis** confirms these qualitative impressions: series from the same domain are much closer to each other than to series from the other domain, meaning that the simulator effectively encodes domain-specific structure.

Pros and cons of the tool

Pros	Cons
• The tool is **simple and intuitive**: parameters correspond to interpretable concepts (trend, noise, seasonal components, peaks and valleys). • It allows the generation of **families of time series** with controlled stylised properties, which is very useful for teaching, prototyping and stress-testing. • Configurations can be saved and re-used, ensuring **reproducibility** of simulations.	• There is limited control over some crucial features, notably the **phase** of seasonal components and more realistic **volatility dynamics**. • Complex structures such as **regime-switching volatility** or non-linear dependencies cannot be modelled directly; they must be approximated using ad-hoc combinations of peaks, valleys and trend randomness. • Parameter tuning is largely based on **trial and error** rather than systematic estimation from data. • The tool does not include built-in **smoothing**, so replicating smoothed empirical figures requires external post-processing.

Overall assessment

In conclusion, the Time Series Random Configurator is well suited to generating **stylised, domain-specific time series** that resemble real data at a qualitative level. It captures the essential differences between a crisis-driven financial index and a strongly seasonal hotel occupancy series, and the distance analysis demonstrates that these two domains form distinct clusters. However, the tool is not designed for exact quantitative replication of empirical data; for that purpose, more formal time-series models and estimation techniques would be required.