perplexity

Geração de Datas Sintéticas com NumPy + Pandas para Charts Plotly e Dashboards Dash

Antes de mergulhar nos detalhes, eis a ideia-força: **pandas + NumPy fornecem todas as engrenagens necessárias para criar séries temporais artificiais que imitam tendências, sazonalidades, anomalias e correlações; combinadas com Plotly e Dash, elas alimentam gráficos interativos e painéis capazes de prototipar cenários complexos sem depender de dados reais**[1] [2].

1. Fundamentos de datas em NumPy / pandas

Pandas abraça o tipo `numpy.datetime64`, oferecendo classes especializadas (`DatetimeIndex`, `PeriodIndex`) e métodos rápidos como `pd.date_range`, `pd.bdate_range` e `to_datetime`[1] [3]. Esses blocos geram sequências de instantes em qualquer resolução – de nanossegundos a anos – respeitando calendários, fusos e feriados.

1.1 Geração direta de calendários

- Diário (`freq='D'`) ou útil (`freq='B'`) para simulações de produção.
- Horário (`'H'`) e por-minuto (`'T'`) para IoT ou monitoramento.
- "Month-End" (`'M'`) e "Quarter-Start" (`'QS'`) para finanças.
- Frequências customizadas com `weekmask` e `holidays` para turnos irregulares.

Alicerce	Exemplo (3 linhas)
Calendário diário	`pd.date_range('2025-01-01', periods=3, freq='D')`
Dias úteis	`pd.bdate_range('2025-01-01', '2025-01-10')`
Horário	`pd.date_range('2025-06-01', periods=24, freq='H')`

2. Padrões clássicos de séries sintéticas

Uma vez que o eixo temporal existe, basta preencher valores. Seguem quatro padrões universais para demonstrar técnicas de modelagem e visualização.

1. **Tendência linear** – `np.linspace` + ruído gaussiano.
2. **Sazonalidade senoidal** – `np.sin` em torno do ano/dia.
3. **Crescimento exponencial** – `np.exp` para churn cumulativo ou usuários ativos.
4. **Eventos raros** – inserir picos em índices sorteados para emular falhas ou campanhas.

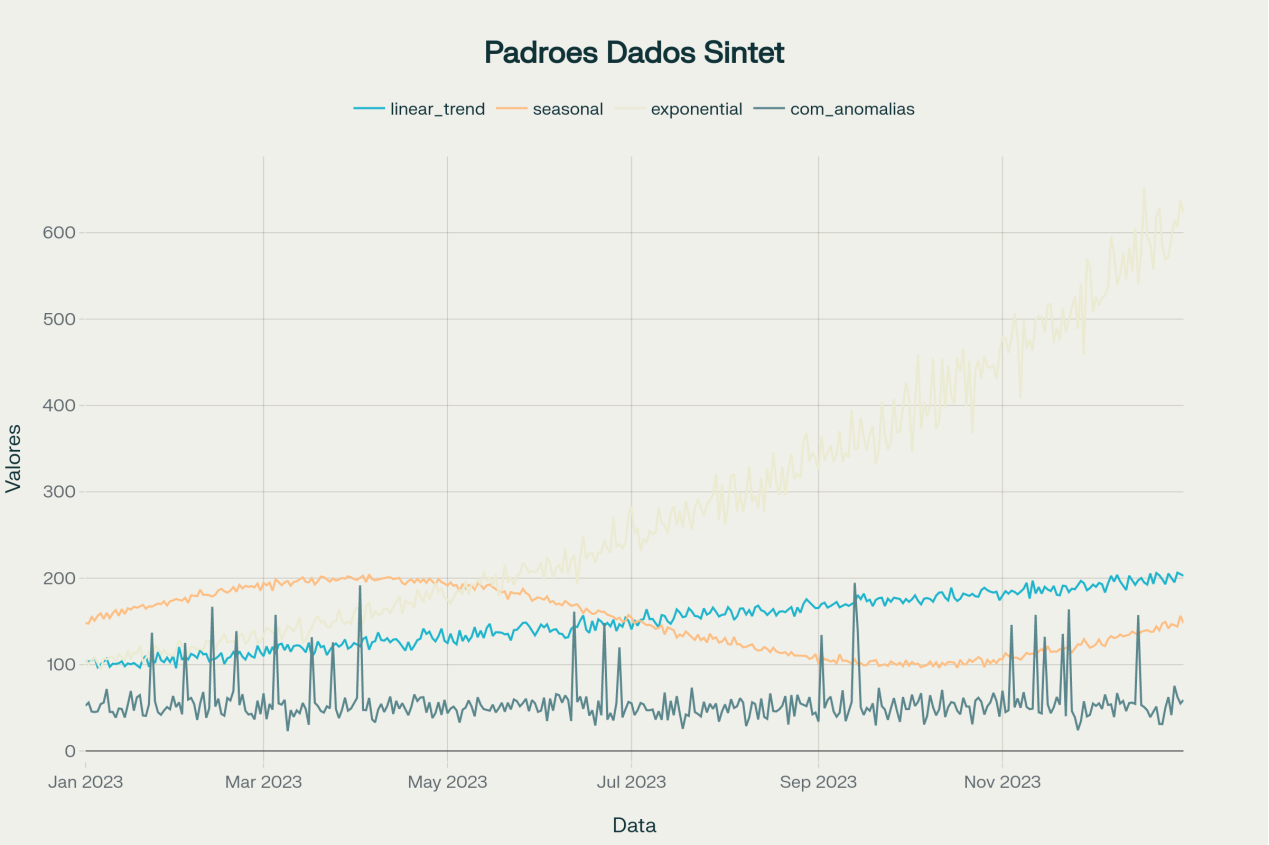


Gráfico mostrando diferentes padrões de séries temporais sintéticas: linear, sazonal, exponencial e com anomalias

Essas curvas foram criadas inteiramente com `pd.date_range` e vetores NumPy (ver código no guia prático). O gráfico evidencia como cada padrão responde a um mesmo calendário, recurso útil para testar algoritmos de suavização ou detecção de anomalias[4].

3. Combinações avançadas

3.1 Múltiplas sazonalidades sobrepostas

Processos energéticos misturam ciclos diurnos, semanais e mensais. A soma de três senos com períodos 24 h, 168 h e 720 h gera um traçado realista; `np.random.normal` adiciona ruído branco, e `np.cumsum` produz ruído “cor-de-rosa” suave.

3.2 Séries correlacionadas

Para testar modelos multivariados basta criar uma “série base” (random walk) e multiplicar por fatores de correlação $ \rho \in [0.3,0.9] $. O restante do termo é ruído ortogonal de peso $\sqrt{1-\rho^2}$. Essa técnica garante matrizes de covariância controláveis, essenciais em simuladores de risco[5].

3.3 Estruturas com mudança de regime

Concatenar metade com tendência e metade com sazonalidade reproduz choques de mercado ou alteração de receita após campanha[6]. `np.concatenate` e fatias do índice bastam.

4. Do dataframe ao gráfico Plotly

4.1 Multi-linhas em um clique

`plotly.express.line` aceita listas de colunas (`y=['linear','season']`) e gera legendas automaticamente. A interatividade (zoom, hover) vem de graça e é ideal para inspeção exploratória.

4.2 Scatter enriquecido

Colorir por volatilidade e dimensionar pelo retorno absoluto (via `size=`) revela padrões de liquidez e risco. Incluir `trendline='ols'` exibe correlação imediatamente – suporte valioso em prototipagem de estratégias[7].

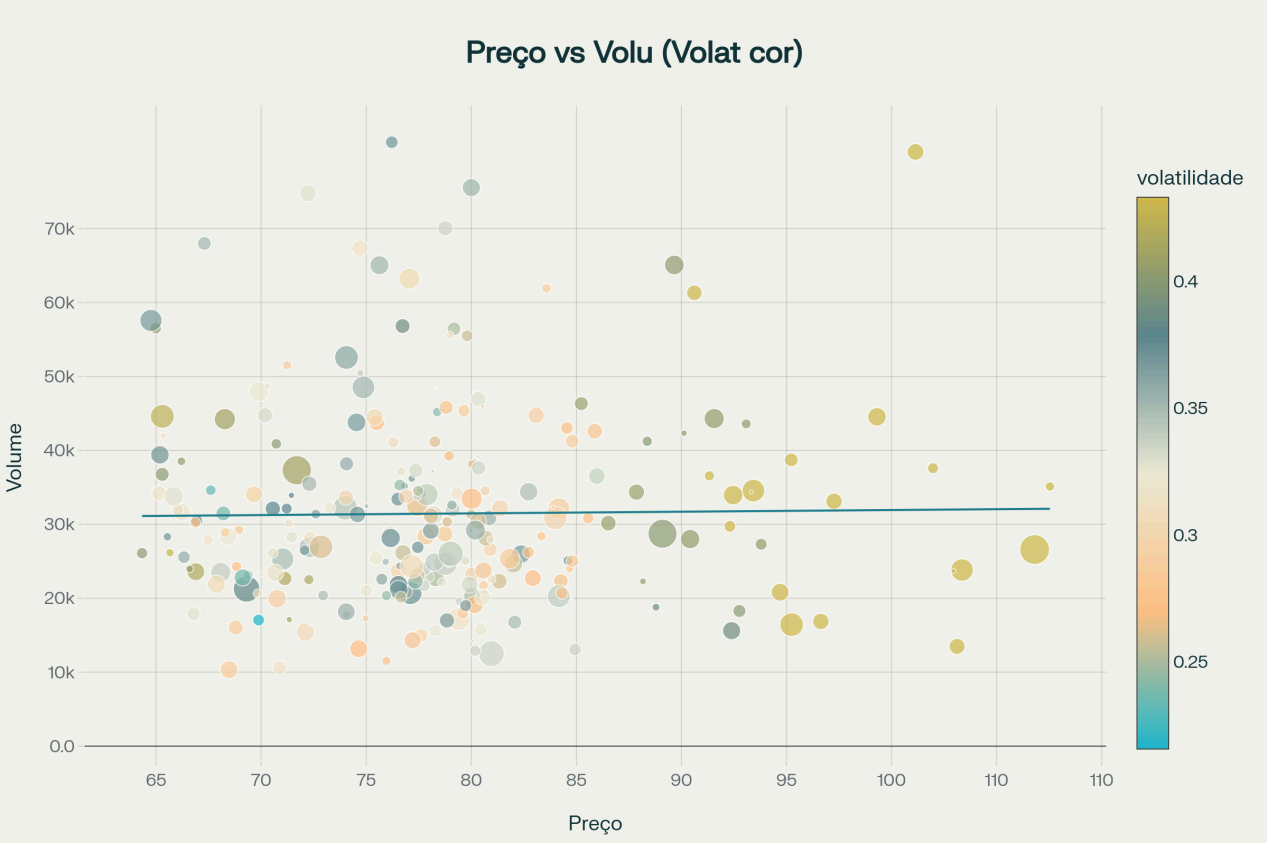


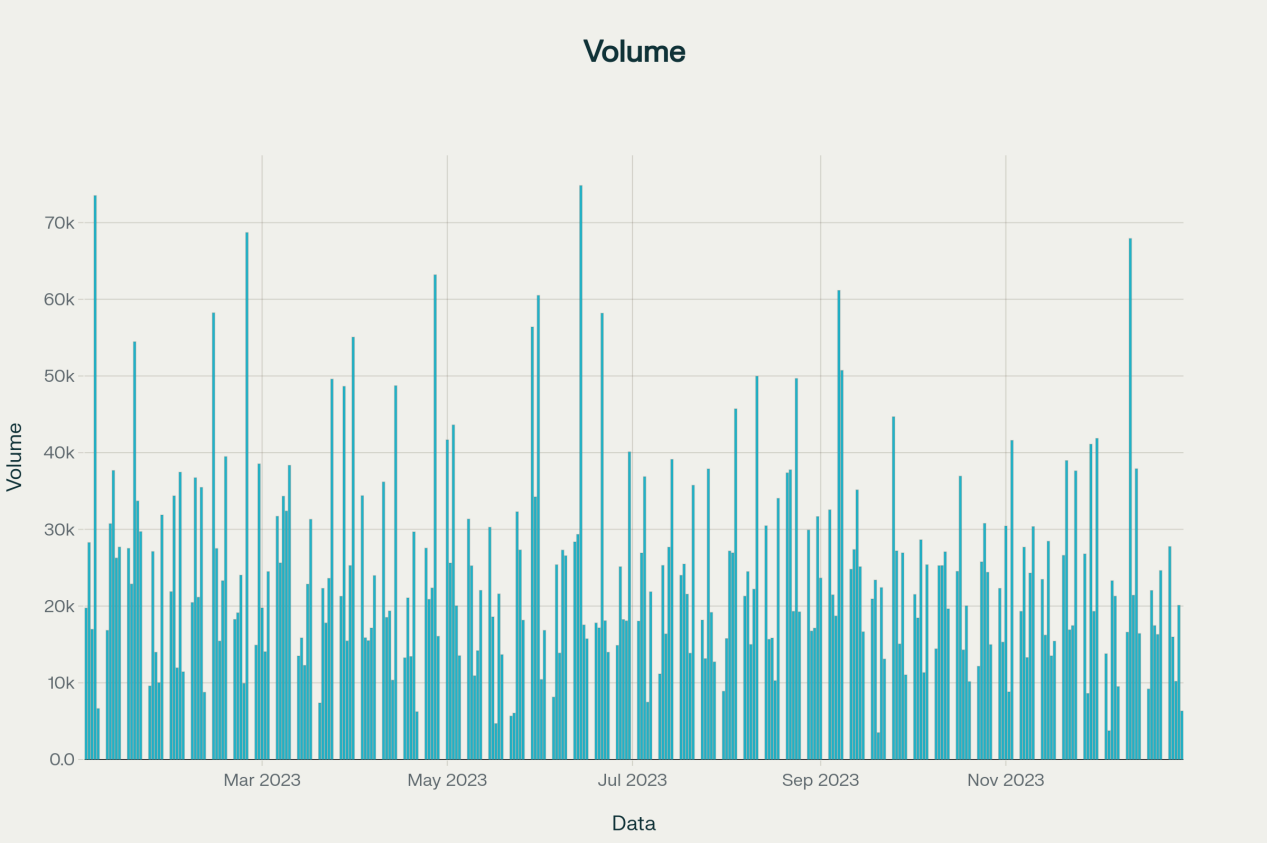
Gráfico de dispersão mostrando correlação entre preço e volume, colorido por volatilidade

4.3 Dashboards empilhados

Com `plotly.subplots` empilhamos três gráficos:

- Linha de preço + média móvel (20 dias).
- Barras de volume negociado.
- Linha de volatilidade anualizada.

Esse painel sintetiza análise técnica e fluxo de ordens em um só canvas.



Dashboard financeiro com três painéis: preços com média móvel, volume de negociação e volatilidade

5. Pipeline Dash completo

O script disponibilizado cria um app Dash com:

- **Controles**: dropdown de série, tipo de gráfico, seletor de período.
- **Callbacks**: filtram o dataframe em tempo real e atualizam gráficos.
- **Estatísticas resumo**: média, mediana, desvio, extremos.
- **Segundo gráfico**: consumo horário com múltiplas sazonalidades.

Basta executar `python dashboard_exemplo.py` para abrir o painel no navegador. Ele ilustra melhores práticas de organização (layout em colunas, tema claro) e mostra como funções de geração podem ser acopladas ao front-end.

6. Estratégias de validação e performance

Desafio	Boa prática
Reprodutibilidade	`np.random.seed(42)` no topo do módulo
Volume massivo ($\geq 10^7$ pontos)	Use tipos `float32` e `Categorical`, habilite **chunking** em Dash (`dcc.Graph(figure, mathjax=False)`)
Feriados nacionais	`CustomBusinessDay(calendar='Brazil/ANBIMA')`
Fusos simultâneos	Armazene UTC no backend e use `tz_convert` na camada de visualização
Qualidade estatística	Compare momento de ordem k (média, variança) entre real e sintético usando `scipy.stats`

7. Bibliotecas que vão além do NumPy / pandas

- **Synthcity** – coleção de GANs, VAEs e copulas para séries de saúde e finanças[8].
- **TSGM** – framework unificado para geração/avaliação de time-series sintéticas[9].
- **StyleTime** – transfere "estilo" estatístico de uma série para outra[10].
- **TarDiff** – difusão guiada por influência para EHRs[11].

Essas ferramentas cuidam de privacidade, preservação de correlações de alto nível e métricas de utilidade, caso você precise de sintéticos para produção.

8. Guia de bolso pronto para uso

Um resumo de todas as receitas de código – do calendário básico ao dashboard mínimo – foi compilado em **guia_datas_sinteticas.md**.

Conclusão

Criar datas e séries temporais sintéticas no ecossistema Python é um processo de **três passos**:

1. **Calendário** – `pd.date_range` / `bdate_range` para a malha temporal correta.
2. **Valores** – combine trend + season + ruído + eventos especiais com NumPy.
3. **Visualização** – use Plotly para protótipos rápidos; promova a produção em Dash.

Com essas peças você consegue desde provas-de-conceito até bancos de teste completos, acelerando P&D sem esbarrar em restrições de dados sensíveis[6] [5].

⁂

1. https://arxiv.org/pdf/2006.10256.pdf
2. https://arxiv.org/abs/2305.18811
3. http://arxiv.org/pdf/2207.05810.pdf
4. https://ieeexplore.ieee.org/document/9973221/
5. https://onlinelibrary.wiley.com/doi/10.1002/er.5115
6. https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-024-02427-0

7. https://ieeexplore.ieee.org/document/9492267/
8. https://arxiv.org/pdf/2301.07573.pdf
9. https://arxiv.org/html/2305.11567v2
10. https://dl.acm.org/doi/10.1145/3533271.3561772
11. https://arxiv.org/abs/2504.17613