Autor: Felipe Rodrigo de Moraes Mendes

Título: Open source software for budget forecasting

# Resumo

Demandas profissionais e acadêmicas para a aplicação de métodos de previsão podem não serem acompanhadas da disponibilidade de ferramentas de *softwares* comerciais para esse fim, como o SAS, Crystal Ball e Minitab. Esse texto argumenta que a utilização da linguagem de programação Python, um *software* livre, pode ser uma alternativa que alia a flexibilidade de uma linguagem de programação com uma abordagem intuitiva. Como estudo de caso utilizou-se dados da execução orçamentária da Universidade Federal de Itajubá, consumidas por meio de uma *application programing interface* (API), publicamente disponível no Portal Transparência, do Governo Federal. A aplicação do método de Holt-Winters foi feita para quatro cenários, considerando as quatro combinações possíveis entre ajustes na tendência e na sazonalidade, aditivas ou multiplicativas. O cenário com tendência e sazonalidade multiplicativa possui menor *root mean squared error* e o menor *mean absolut error*.

**Palavras-chave**: orçamento federal; previsão; python.

# Introdução

Organizações de todos os ramos de atuação e tamanho podem se beneficiar do uso de técnicas de previsão para planejar seu orçamento. Isso não é diferente para instituições de ensino superior públicas brasileiras. No presente trabalho, dados orçamentários e de execução da despesa da Universidade Federal de Itajubá são utilizados para como estudo de caso na aplicação da linguagem de programação Python para previsão de séries temporais.

Considerando previsão orçamentária, dois aspectos devem ser observados, conforme exposto por Williams e Calabrese (2016) e Penner (2019): o período de previsão e se a previsão terá como objeto receitas ou despesas. Williams e Calabrese (2016) informam que os usos de previsão aplicadas no orçamento podem ser aplicados no processo para três diferentes períodos: (1) existe a previsão intra ano, chamada de *current year*, onde se utilizam dados para previsão dentro do ano corrente; (2) a previsão para o ano orçamentário, chamado de *budget year*, onde se busca prever o orçamento do ano seguinte ao ano corrente e (3) previsão para anos posteriores, chamado de *out years*, onde a previsão ocorre a partir do final do ano orçamentário. Ainda segundo Williams e Calabrese (2016), os mesmos autores, em cada período pode-se focar em prever dados referentes às diversas formas de receitas do órgão ou dados referentes à previsão das despesas a serem executadas.

Penner (2019) relativiza a ação de prever apontando que previsões de longo prazo podem gerar mais erros quando comparadas a previsões de curto prazo (até um ano). Contudo, o autor afirma que previsões não precisam estar corretas para apresentar valor e que identificar a direção na qual receitas e despesas estão caminhando já pode ser útil (PENNER, 2019, p. 116).

No contexto prático, Williams e Calabrese (2016) apontam que o uso de *softwares* computacionais facilitam as aplicações práticas de métodos para previsão orçamentária, pois as complexas modelagens matemáticas e os seus respectivos cálculos são realizados pela máquina e o usuário opera o sistema por meio de interfaces gráficas. Contudo, o uso de pacotes de *software* comerciais como o SAS, o Crystal Ball e o Minitab tem alto custo e podem não estar disponíveis na realidade do profissional. Os autores apontam como uma alternativa o uso do *software* estatístico “R”. Contudo, esse *software* apresenta alta curva de aprendizado e demanda conhecimento técnico específico do usuário, uma vez que os modelos e interações matemáticas devem ser previamente programadas na linguagem própria do programa.

Como uma solução, que é *open source* como o *software* “R”mas que apresenta curva de aprendizado menor é a linguagem de programação Python, que conta com pacotes específicos para a análise de séries temporais, previsão e diagramação de gráficos. Nesse sentido, Garita (2021) apresenta três principais razões para o uso da linguagem Python: (1) a linguagem é de fácil uso e a curva de aprendizado é mais achatada (comparada com o “R”) e a forma de programação é intuitiva; (2) o Python apresenta um recurso chamado de *Jupyter Notebooks*, que confluem um editor de código interativo que permite a execução de linhas de código individualmente, com funcionalidades de *markdown*, para anotações em texto, e (3) existem bibliotecas dedicadas a diagramação de gráficos, como a biblioteca *Matplotlib*.

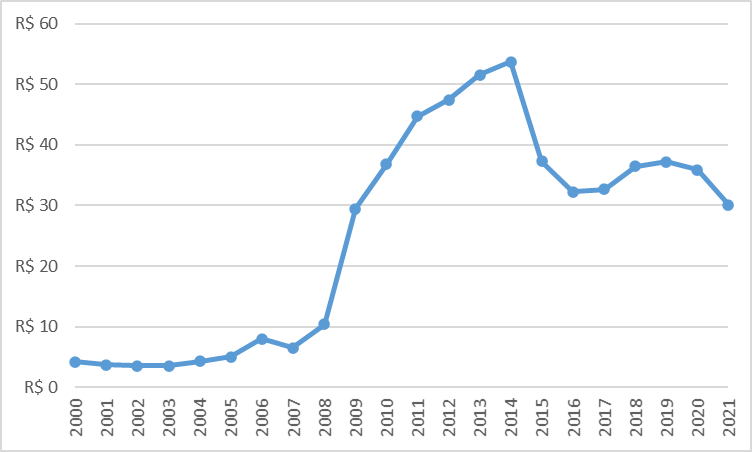
Como forma de demonstrar a aplicação da linguagem Python no contexto da previsão de dados orçamentários, recorreu-se à busca por dados abertos do orçamento e da execução de despesas da Universidade Federal de Itajubá (Unifei). Após essa introdução, a próxima seção mostra as fontes dos dados e os métodos aplicados para sua coleta e análise. Na terceira seção, os resultados práticos da aplicação do método preditivo são mostrados. Na quarta seção, apresentam-se as conclusões.

# Caminhos metodológicos

Com os objetivos de determinar qual fonte de dados utilizar para obter as séries temporais e escolher o método adequado para sua análise, buscou-se dados de fontes governamentais, uma vez que o governo brasileiro disponibiliza acesso público a dados orçamentários e de realização da despesa de seus órgãos.

Primeiramente recorreu-se ao Sistema Integrado de Planejamento e Orçamento (SIOP), disponível em <https://www.siop.gov.br/>. Contudo, os dados disponíveis são limitados e mostram uma janela temporal anual com apenas 22 observações, referentes aos anos de 2000 a 2021. O resumo gráfico desses dados encontra-se na Figura 1.

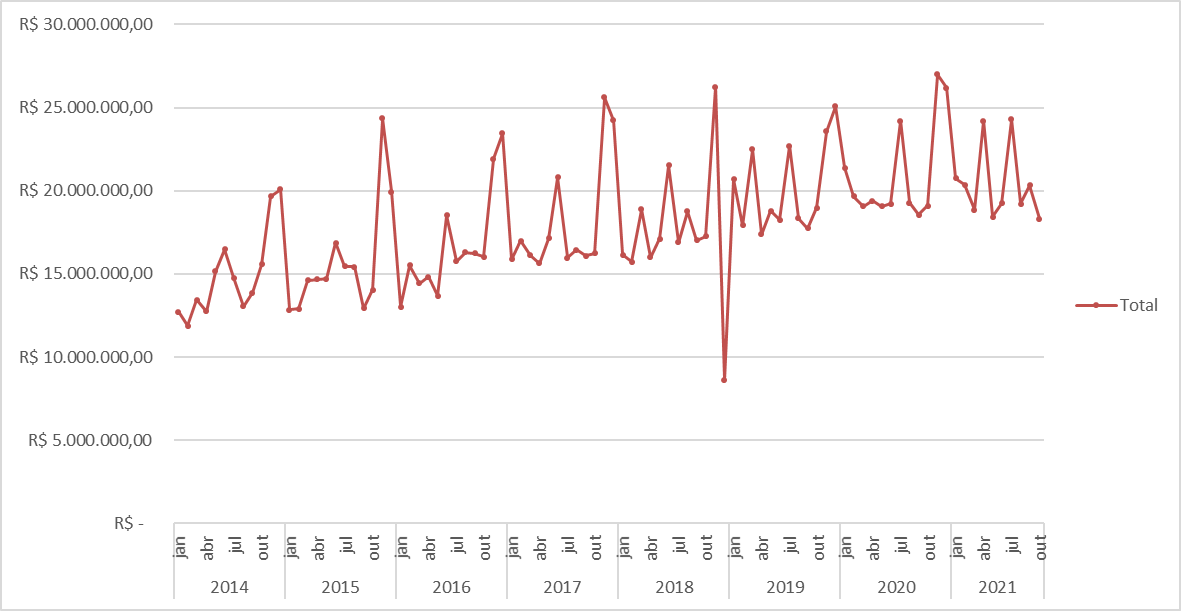
Figura 1 - Orçamento da Unifei em milhões de reais (verbas discricionárias)



Fonte: contribuição do autor, com base em consulta realizada ao SIOP)

Para contornar essa limitação dos dados, recorreu-se a *application programming interface* ou API, do Portal Transparência, disponível em <http://www.portaltransparencia.gov.br/>, onde foi possível coletar dados da execução de despesa da Unifei. Essa nova fonte de dados apresenta evolução mensal em 94 pontos de dados, referentes a janeiro de 2014 até outubro de 2021. O resumo gráfico desses dados encontra-se na Figura 2.

Figura 2 - Execução da Despesa Mensal - Jan. de 2014 até out. de 2021 da Unifei



Fonte: contribuição do autor.

A análise do gráfico da Figura aponta duas características desse conjunto de dados: (1) ele apresenta tendência geral de crescimento, pois a amplitude de variação do primeiro ano varia, aproximadamente, entre R$ 12 milhões até R$ 20 milhões, em 2014, e varia, aproximadamente, entre R$ 20 milhões até R$ 24 milhões, em 2021; (2) apresenta sazonalidade, com expressiva alta na execução orçamentária nos meses de novembro e dezembro de cada ano.

Comparando-se os dados resultantes das consultas feitas, optou-se por se utilizar os dados do Portal Transparência por duas razões: (1) apresenta mais dados, 94 observações comparadas com as 22 observações do SIOP e (2) apresenta características de tendência e sazonalidade.

Escolhida a fonte de dados, partiu-se para a definição de qual método de previsão utilizar. Consultando referências especializadas, as três fontes foram unânimes ao apontarem o método de Holt-Winter de alisamento exponencial (MILLS, 2019, p. 153; MONTGOMERY; JENNINGS; KULAHCI, 2015, p. 277; SHMUELI; LICHTENDAHL JR., 2016, p. 95). Esse método permite que padrões no nível, na tendência e na sazonalidade mudem no tempo. Ademais, esse método admite componentes aditivos e componentes multiplicativos, conforme exemplificados no Quadro 1. Considerando que existem quatro combinações possíveis entre tendências e sazonalidades aditivas e multiplicativas, quatro cenários foram criados, conforme foram destacados na seção de resultados.

Para as métricas de erro, serão utilizadas a *mean absolute error* (MAE) e a *root mean squared error* (RMSE), como visto em Shmueli e Lichtendahl Jr. (2016, p. 55) e resumidos no Quadro 2.

Quadro 1 - Exemplos das formulações de modelos aditivos e multiplicativos.

| **Equações** | **Tendência aditiva e sazonalidade multiplicativa** | **Tendência e sazonalidade aditivas** |
| --- | --- | --- |
| Geral |  |  |
| Ajuste do nível |  |  |
| Ajuste da tendência |  |  |
| Ajuste da sazonalidade |  |  |

Fonte: resumo proposto pelo autor, adaptado de Shmueli e Lichtendahl Jr. (2016).

As variáveis descritas no Quadro 1 significam:

* : previsão para o próximo período *k* no período atual *t;*
* : nível, do inglês *level*, que descreve o valor médio da série no período atual *t;*
* : tendência, do inglês *trend*, descreve a mudança de valores no período atual *t* comparado com períodos anteriores;
* : sazonalidade, do inglês *seasonality*, que descreve um comportamento cíclico, no período *t*, que pode ser observado repetidas vezes na série;
* : constantes de alisamento ligadas, respectivamente, ao nível, tendência e sazonalidade
* *M*: temporadas da sazonalidade.

Quadro 2 - Resumo das métricas de erro que foram utilizadas.

| **Métrica** | **Formulação** | **Implicação prática** |
| --- | --- | --- |
| *Forecast error* |  | Diferença entre o valor atual e o valor da previsão. |
| *Mean absolute error* |  | Essa métrica fornece a magnitude do erro absoluto médio. |
| *Root mean squared error* |  | Raiz quadrada do erro médio ao quadrado. Essa medida apresenta as mesmas unidades que a série temporal original. |

Fonte: resumo proposto pelo autor, adaptado de Shmueli e Lichtendahl Jr. (2016).

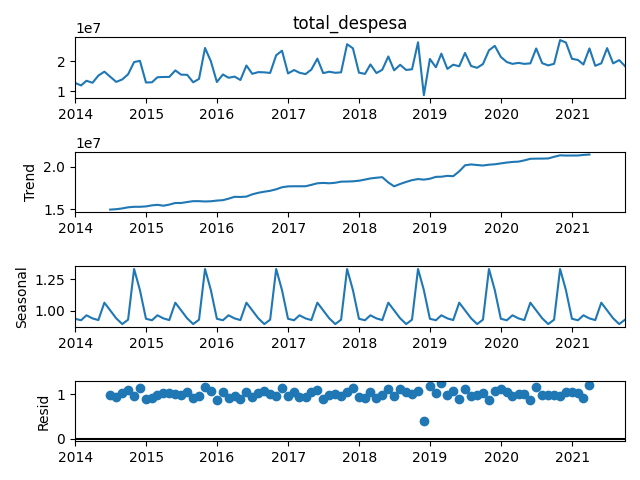
Definida qual a fonte de dados utilizar e qual método aplicar nos dados, é necessário definir como se dará a aplicação em Python. Para tal, as seguintes bibliotecas foram utilizadas: (1) *pandas*, utilizada para leitura dos dados e sua organização em formato tabular; (2) *Matplotlib*, utilizada para a geração de gráficos 2D; (3) *statsmodels*, utilizada para decompor a série temporal e realizar o alisamento exponencial de Holt Winters e (4) *scikit-learn*, utilizado para quantificar as métricas de erro do modelo. Os dados e os *scripts* utilizados para gerar os resultados estão publicamente disponíveis em <https://bit.ly/PQM14_PrevisaoUnifei>.

Na próxima seção os resultados da aplicação da linguagem Python aos dados serão mostrados.

# Resultados

A biblioteca *statsmodels* possui recurso que permite decompor a série temporal e visualizar graficamente seus componentes, conforme Figura 3. É possível concluir que os dados apresentam tendência positiva crescente e sazonalidade, com picos menores, registrados nos meses de junho de cada ano e picos maiores, registrados nos meses de novembro de cada ano.

Figura 3 - decomposição da série temporal utilizando o modelo multiplicativo



Fonte: contribuição do autor.

Na condução dos experimentos, representou-se graficamente os dados utilizados como treinamento, os dados de teste e os dados da previsão. O resumo gráfico está presente na Figura 4.

Figura 4 - Resumo gráfico dos quatro cenários

|  |  |
| --- | --- |
|  |  |

Fonte: contribuição do autor.

Os dados utilizados para o treinamento da ferramenta são referentes a 84 pontos de dados mensais, referentes aos anos de 2014 a 2020, e os testes foram realizados com 10 pontos de dados do ano de 2021, de janeiro a outubro.

Uma inspeção gráfica dos quatro cenários não mostra grandes diferenças comparativas. Contudo, ao se calcular as métricas de MAE e RMSE, é possível identificar que o menor MAE foi registrado pela combinação entre tendência aditiva e sazonalidade multiplicativa e o menor RMSE está relacionado com a combinação entre tendência e sazonalidade aditivas. Todos os MAE e MSE dos quatro cenários estão registrados no Quadro 2.

Quadro 2 - Comparação tabular entre as métricas de erro de cada cenário

| **Tipo Tendência** | **Tipo Sazonalidade** | **MAE** | **RMSE** |
| --- | --- | --- | --- |
| Aditiva | Aditiva | 2.199.355,29 | 2.595.644,75 |
| Aditiva | Multiplicativa | 2.220.480,11 | 2.821.949,63 |
| Multiplicativa | Multiplicativa | **2.195.483,64** | **2.579.724,43** |
| Multiplicativa | Aditiva | 2.201.811,25 | 2.599.131,47 |

Fonte: Contribuição do autor.

Nota: Valores em negrito representam os menores erros dentro dos quatro cenários.

Considerando possíveis aplicações práticas do uso dessas previsões para o planejamento orçamentário e o conceito de previsão para o *buget year* trazido por Williams e Calabrese (2016) e utilizando modelo com tendência e sazonalidade multiplicativa, seria possível prever a execução da despesa de 2021 com erro MAE de R$ 2.195.483,64 e RMSE de R$ 2.579.724,43. Isso fornece ao gestor dados para suporte à decisão com o objetivo de melhorar os processos de planejamento institucional.

# Conclusão

A aplicação da linguagem Python para a previsão de séries temporais no contexto da previsão orçamentária é possível. Apesar de não possuir interface gráfica, a utilização do Python é intuitiva e apresenta bons resultados. Apesar de o autor já ter experiência com outras bibliotecas do Python, ele não havia trabalhado com previsão de séries temporais. Com base na experiência com Python e com experiências prévias do autor com R e Minitab, é possível dizer que a curva de aprendizado no Python é mais suave.

No campo das contribuições práticas, os resultados dessa previsão podem ser úteis para gestores programarem o ano orçamentário de seu órgão baseado em métodos quantitativos e racionais.

Como uma limitação, as bibliotecas utilizadas, principalmente a *statsmodels* apresentam situações classificadas como “caixas pretas” onde o usuário solicita ao código que determinada operação seja conduzida, mas as rotinas internas que efetivamente realizam os cálculos não são facilmente identificáveis e exigem um conhecimento mais avançado para conhecê-las, como a depuração do código fonte.

Outra limitação e consequente sugestão de pesquisa futura é a comparação do uso de Python com outras ferramentas, como a linguagem R, SAS e Minitab. As métricas de MAE e RMSE aqui apresentadas podem servir como benchmarking e fontes de comparação.

# Referências

BROWNLEE, Jason. **Introduction to time series forecasting with python**: how to prepare data and develop models to predict the future. Jason Brownlee, 2019.

GARITA, Mauricio. Why Python?. In: **Applied Quantitative Finance**. Palgrave Pivot, Cham, 2021. p. 1-17.

MILLS, Terence C. **Applied time series analysis: A practical guide to modeling and forecasting**. Academic press, 2019.

MONTGOMERY, Douglas C.; JENNINGS, Cheryl L.; KULAHCI, Murat. **Introduction to time series analysis and forecasting**. John Wiley & Sons, 2nd ed. 2015.

PENNER, Rudolph. **The Reliability of Long-Run Budget Projections**. In: WILLIAMS, Daniel; CALABRESE, Thad. The Palgrave handbook of government budget forecasting. Palgrave Macmillan, 2019.

SHMUELI, Galit; LICHTENDAHL JR., Kenneth C. Practical time series forecasting with R: A hands-on guide. Axelrod Schnall Publishers, 2nd ed. 2016.

SIEBERT, Julien; GROB, Janek; SCHROTH, Christof. A systematic review of Python packages for time series analysis. **arXiv preprint** arXiv:2104.07406, 2021.

WILLIAMS, Daniel W.; CALABRESE, Thad D. The status of budget forecasting. **Journal of Public and**

**Nonprofit Affairs**, v. 2, n. 2, p. 127-160, 2016.