

Relatório - Teste M|o|S Capital

Cálculo de Beta de ações

A proposta do projeto é realizar o cálculo de Beta de 5 ações SUZB3, PETR4, ITUB4, LREN3 e WEGE3 em períodos fixos de 1, 3 e 5 anos, utilizando os índices Ibovespa.

O teste foi realizado em dois caminhos:

- **VBA:** Criar uma macro que calcule automaticamente o beta para 1, 3 e 5 anos das 5 ações, atualize os gráficos no Excel, e inclua um botão para refazer os cálculos e gráficos ao ser clicado.
- **Python:** Desenvolver um script que importe dados de CSV ou Excel, calcule os betas para 1, 3 e 5 anos, e exporte os resultados e gráficos prontos para uso em um novo arquivo Excel.

Para cada ação, serão gerados:

- Python
 - Um Excel com os resultados, sem gráfico.
 - Um Excel com os resultados e gráficos de análise.
 - Um gráfico utilizando a biblioteca Python Matplotlib.
- VBA
 - Um .xlms com os resultados e graficos

O objetivo é comparar os formatos e a interpretação dos dados em diferentes representações. A análise do beta é uma medida de sensibilidade de um ativo em relação ao comportamento de um índice. No projeto foi utilizado os índices da Ibovespa. O objetivo é medir a volatilidade de uma ação, sendo uma das principais ferramentas para investidores que desejam entender o risco associado a cada ativo.

- Excel Sem Gráfico: Este arquivo fornecerá uma visão clara e concisa dos dados calculados, permitindo uma rápida consulta aos valores de beta ao longo dos períodos analisados.
- Excel Com Gráfico: Este arquivo incluirá gráficos que ilustram a relação entre o retorno das ações e do índice. Gráficos de linhas serão utilizados para destacar tendências e variações no beta ao longo do

período analisado. Para automatização desses gráficos foi utilizado linguagem de programação Python e VBA

- Gráfico Utilizando Matplotlib: Este gráfico será gerado em Python usando a biblioteca Matplotlib e terá como objetivo representar graficamente o comportamento do beta ao longo do tempo.

Ao final da análise, será possível tirar conclusões sobre a sensibilidade das ações escolhidas em relação às variações do mercado, ajudando os investidores a tomar decisões mais informadas sobre suas carteiras de investimento. O uso de diferentes formatos de apresentação permitirá não apenas uma compreensão profunda dos dados, mas também uma comparação entre as ferramentas utilizadas, destacando suas respectivas vantagens e limitações, e contribuindo para uma escolha mais adequada das metodologias de análise em futuras avaliações de risco e desempenho de ativos.

Dados:

- Data inicial dos dados para análise: 01/01/2019 - 23/08/2024^[1].
 - Fontes: Investing: <https://br.investing.com> (SUZB3, PETR4, ITUB4, LREN, 3WEGE3, Ibovespa); Yahoo Finance - Finance's API: <https://pypi.org>

Python:

Desenvolvimento

O desenvolvimento do **analiseBeta** foi realizado utilizando a linguagem de programação Python. O projeto foi estruturado em diferentes pacotes para separar as responsabilidades e facilitar a manutenção e extensibilidade do código. (csv, model, view, vba)

Para o gerenciamento de dependências e ambientes foi utilizado a biblioteca **Poetry** e para facilitar o controle de versões e desenvolvimento colaborativo o projeto está disponível no Github^[2].

Decisões de funcionalidades:

O Sistema de Cálculo de Beta possui as seguintes funcionalidades principais:

1. Model:

- Permite faxinar os dados: **processar_dados_csv()**
- Buscar dados de mercado no Yahoo Finance: **buscar_dados_yahoo()**

- Calcula o Beta de uma ação: `calcular_beta()`

2. View:

- Exporta os retornos e o beta calculado para um arquivo excel: `exportar_para_excel()`
- Exporta os retornos e o beta para o excel automatizando a criação dos gráficos:
`exportar_para_excel_com_grafico()`
- Cria gráficos com a biblioteca `matplotlib`: `gerar_graficos()`

3. Source:

- Organiza todo o sistema de automatização, facilitando a manipulação dos dados em um período determinado: `gerar_graficos_anos()`

Uso

Exemplo de chamada da função `exportar_para_excel_com_grafico`

```
from gerar_graficos_anos import calcular_e_gerar_graficos

# CÁLCULO DO BETA
# SUZB3
caminho_base_suzb3 = 'src/csv/SUZB3.csv'
calcular_e_gerar_graficos(caminho_base_suzb3, 'SUZB3', 1, '2023-05-13', '2024-05-03', 'doc/SUZB3sg_1ano.xlsx')
calcular_e_gerar_graficos(caminho_base_suzb3, 'SUZB3', 3, '2021-07-31', '2024-07-31', 'doc/SUZB3sg_3anos.xlsx')
calcular_e_gerar_graficos(caminho_base_suzb3, 'SUZB3', 5, '2019-08-28', '2024-08-28', 'doc/SUZB3sg_5anos.xlsx')
```

VBA

Também foi utilizada uma macro em VBA para auxiliar na manipulação e formatação dos dados no Excel.

Foi necessário fazer algumas alterações manualmente:

- Conversão das células para number e percent
- Converter data para Date ('yyyy/mm/dd')

- Inclui uma nova aba para os índices da ibovespa
- Deixar datas em ordem crescente

Foram implementadas essas rotinas:

- Sub `calcularBeta()`
- Sub `GerarGraficoPorPeriodo()`

Para gerar automaticamente os 3 gráficos diferentes foram criadas em outro módulo:

- Sub `GerarGraficoPeriodoEspecifico1anos()`
- Sub `GerarGraficoPeriodoEspecifico3anos()`
- Sub `GerarGraficoPeriodoEspecifico4anos()`

Calculo do BETA:

O cálculo do beta de um ativo envolve vários passos^[3]:

- Coletar dados históricos dos retornos do ativo e do índice de referência.
- Calcular os retornos do ativo e do índice.
- Calcular a covariância entre os retornos do ativo e do índice.
- Calcular a variância do índice (*Ibovespa*).

Implementação do cálculo no Python:

A função `calcular_beta(dados_mercado, dados_processados)` executa os seguintes passos:

- Cálculo dos Retornos Diários:

Os dados são ajustados para definir `'Data'` como o índice do DataFrame.

Utiliza-se a função `pct_change()` para calcular os retornos diários da coluna `'Último'`, removendo valores nulos com `dropna()`. Isso garante que apenas dados válidos sejam considerados.

- Combinação dos Dados:

Os retornos da ação e do mercado são combinados em um DataFrame, permitindo cálculos de covariância e variância.

- Cálculo da Variância do Mercado:

A variância dos retornos do mercado é calculada utilizando `np.var()`. Se a variância for zero ou se houver valores nulos, um erro é levantado, garantindo a integridade dos cálculos.

- Cálculo da Covariância e do Beta:

A covariância entre os retornos da ação e do mercado é calculada usando `np.cov()`, e o Beta é obtido pela divisão da covariância pela variância do mercado.

Implementação em VBA

- Cálculo dos Retornos Diários:

Os retornos são obtidos da coluna de variação percentual das ações e do Ibovespa, para convertê-los em formato decimal.

- Inicialização de Variáveis:

Arrays são criados para armazenar os retornos, e variáveis de soma são inicializadas para o cálculo da média.

- Cálculo das Médias:

As médias dos retornos são calculadas somando os retornos e dividindo pelo número total de períodos.

- Cálculo da Covariância e Variância:

A covariância é calculada somando os produtos das diferenças dos retornos em relação às médias. A variância do Ibovespa é calculada somando os quadrados das diferenças.

- Cálculo do Beta:

O Beta é calculado ao final, e os resultados são armazenados em uma nova aba chamada “Beta”.

Estrutura do Projeto

O projeto foi estruturado em diferentes pacotes para separar as responsabilidades e facilitar a manutenção e extensibilidade do código.

- `csv` : Contém todos os arquivos .csv
- `model` : Códigos referente a modelagem do projeto
- `view` : Códigos referente todas as visualizações dos gráficos
- `vba` : Contém todas as planilhas `.xlsm` e os códigos responsáveis pelas macros.

Existe uma pasta doc (os resultados gerados com Python).

Para utilizar do sistema compile o arquivo Main.

Análise e Conclusões

Após a análise dos betas das ações SUZB3, PETR4, ITUB4, LREN3 e WEGE3 ao longo dos períodos determinados, foram identificados os seguintes padrões:

- **SUZB3** (Suzano S.A.): Observou-se uma alta abrupta nos retornos em maio de 2023, seguida por instabilidade entre setembro de 2023 e maio de 2024, com variações significativas. O ano de 2020 também apresentou elevada instabilidade.
- **PETR4** (Petrobras): Houve uma queda acentuada em maio de 2023, indicando alta volatilidade. Apesar disso, a maior parte do período analisado mostrou estabilidade, destacando-se uma alta significativa no início de 2022. Entre 2020 e 2021, verificou-se instabilidade, com uma recuperação notável entre 2022 e 2023.
- **ITUB4** (Itaú Unibanco): Registrou-se uma queda abrupta em setembro de 2023, sugerindo elevada instabilidade. No entanto, o restante do período apresentou relativa estabilidade, com destaque para uma alta significativa no início de 2021. O ano de 2020 mostrou-se particularmente instável.
- **LREN3** (Lojas Renner): Apresentou alta instabilidade ao longo de todo o ano, com variações consideráveis nos retornos. Os anos anteriores também mostraram volatilidade, especialmente entre janeiro e maio de 2021 e 2022. O ano de 2020 destacou-se pela instabilidade.
- **WEGE3** (WEG): Houve uma alta abrupta em novembro de 2023. Entre setembro de 2023 e janeiro de 2024, verificou-se uma alta significativa, seguida por uma queda entre maio e setembro de 2024. O ano de 2020 apresentou elevada instabilidade.

De maneira geral, todas as ações analisadas mostraram instabilidade significativa no ano de 2020, possivelmente refletindo os impactos da pandemia de COVID-19 no mercado financeiro.

A comparação dos betas ao longo dos diferentes períodos sugere que algumas ações, como SUZB3 e LREN3, são mais sensíveis a curto prazo, enquanto outras, como ITUB4 e WEGE3, apresentam um comportamento mais estável e defensivo ao longo do tempo.

Esses insights podem ajudar os investidores a tomar decisões estratégicas sobre a composição de suas carteiras, equilibrando risco e retorno de acordo com o perfil de volatilidade das ações.

O uso das ferramentas Python e VBA para o cálculo e análise dos betas permitiu uma visão abrangente sobre a volatilidade das ações em relação ao índice Ibovespa. O projeto forneceu diferentes formatos de visualização (tabelas, gráficos no Excel e gráficos gerados por Python), facilitando uma análise detalhada do comportamento das ações em múltiplos períodos.

Além dessas conclusões, existe a possibilidade de desenvolver uma aplicação Windows para interagir com os códigos Python, proporcionando uma interface amigável que automatiza os cálculos e facilita a usabilidade para os usuários.

Links importantes

.ppt : [Apresentação com análises mais detalhadas dos resultados](#)

drive : [Integração de planilhas com .ppt](#)

Referências

1. Google Scholar. (s.d.). *Artigos Acadêmicos sobre Beta*. Disponível em: <https://scholar.google.com>
2. Medium. (2020). *Beta*. Disponível em: <https://www.medium.com>
3. Yahoo Finance - Finance's API. (s.d.). *Yfinance*. Disponível em: <https://pypi.org>

Notas de rodapé:

1. Data que os dados foram colhidos do site. ↩
2. [analiseBeta](#) ↩
3. [Índice Beta](#) ↩