



**Universidade Federal do Pará
Campus Universitário Castanhal
Mestrado Profissional em Matemática em Rede Nacional**

Dissertação de Mestrado

**Investigações de Carteira com o Python
como Proposta no Ensino de Elementos
da Estatística Descritiva**

Hernandes Macedo de Sousa

Castanhal
2023

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

S725i Sousa, Hernandes Macedo de.
Investigações de Carteira com o Python como Proposta no
Ensino de Elementos da Estatística Descritiva / Hernandes Macedo
de Sousa. — 2019.
x,101 f. : il. color.

Orientador(a): Prof. Dr. Valdelírio da Silva E Silva
Dissertação (Mestrado) - Universidade Federal do Pará,
Campus Universitário de Castanhal, Programa de Pós-Graduação
em Matemática em Rede Nacional, Castanhal, 2019.

1. Estatística Descritiva. 2. Modelo de Markowitz. 3.
Índice Sharpe. 4. Python. 5. Tecnologias Digitais de
Informação e Comunicação (TDIC). I. Título.

CDD 310.7

Investigações de Carteira com o Python como Proposta no Ensino de Elementos da Estatística Descritiva

Mestrado

03/2021 – 12/2023

Submissão 22/11/2023

Defesa 30/11/2023

Versão Final 05/12/2023

Universidade Federal do Pará
Campus Universitário Castanhal
Mestrado Profissional em Matemática em Rede Nacional

Hernandes Macedo de Sousa

hernandes.sousa@escola.seduc.pa.gov.br

Mestrando do ProfMat

UFPA-Castanhal

Banca Examinadora

Prof. Valdelírio da Silva e Silva

Orientador

Prof. Dr. João Cláudio Chamma Carvalho

Membro Externo

Prof. Dr. Renato Germano Reis Nunes

Membro do Profmat

Agradecimentos

Agradeço a Deus;

Aos meus familiares mais próximos: minha mãe Maria das Graças; ao meu pai Raimundo Carlos (*in memorian*); irmãs Carla Sousa e Glaucia Sousa; minhas sobrinhas Camila e Paula; minha companheira Fabíola Chaves; e meus dois filhos César e Pedro.

Agradeço aos meus professores: Prof. Dr. Edilberto Oliveira Rozal; Prof. Dr. Arthur da Costa Almeida; Prof^a. Dr^a. Gerlândia de Castro Silva Thijm; Prof. Dr. Nielsen Fernando Lisboa da Silva; Prof. Dr. Samuel Levi Freitas da Luz; e em especial ao Prof. Dr. Valdelírio da Silva e Silva, que foi dedicado e incansável na orientação deste trabalho.

Aos meus amigos de classe: Prof. João Carlos Amador; Prof. Elias Macêdo; e Prof. Elvis Dias.

Enfim, agradeço a todos os que de alguma maneira tenham me ajudado ou estavam torcendo por mim durante esta caminhada.

Resumo

Apresentamos análises baseadas nos retornos diários de ativos de uma carteira de investimento a fim de apresentar elementos de estatística descritiva e de decisão de escolha de pesos do portfólio, mediante uma aproximação do máximo índice Sharpe no modelo de Markowitz. As informações de histórico dos ativos de renda fixa e variável são importados de endereços eletrônicos, e os dados são tratados para extrair informações relevantes para estudos. Tais tarefas são construídas no *Python*, através da biblioteca *Pandas* e cujas visualizações gráficas também fazem uso de outros pacotes. Todos os comandos na confecção para a aproximação são apresentados, com justificativas das sintaxes, e disponibilizados em notebook do *Google Colaboratory*. Comparamos o rendimento da carteira adotada pela aproximação com a obtida por otimização, com intuito de fomentar discussões da estratégia adotada e configurar o trabalho como um proposta de ensino de alguns elementos de estatística, de educação financeira e de lógica de programação.

Palavras-chaves: Estatística Descritiva. Modelo de Markowitz; Índice Sharpe; *Python*; Tecnologias Digitais de Informação e Comunicação (TDIC); Educação Financeira; Lógica de Programação.

Abstract

We present analyzes based on the daily returns of assets in an investment portfolio with the aim of presenting descriptive statistics and decision-making elements for choosing portfolio weights, through an approximation of the maximum Sharpe ratio in the Markowitz model. Historical information about fixed and variable income assets is imported from electronic addresses and the data is processed to extract relevant information for studies. Such tasks are built in *Python*, using the *Pandas* library and whose graphical visualizations also make use of other packages. All commands in creating the approach are presented, with syntax justifications, and made available in a *Google Colaboratory* notebook. We compared the performance of the portfolio adopted by the approximation with that obtained by optimization, with the aim of stimulating discussions about the adopted strategy and configuring the work as a proposal for teaching some elements of statistics, financial education and programming logic.

Keywords: Descriptive Statistics. Markowitz Model. Sharpe Ratio. *Python*. Digital Information and Communication Technologies (DICT). Financial Education. Programming Logic.

Lista de Figuras

2.1	Informações do ativo	26
2.2	Tipos de Gráfico	27
2.3	Gráfico circular	27
2.4	Representação dos ângulos do setor circular	29
2.5	Gráfico de linhas	29
2.6	EQTL3 e MGLU3	35
3.1	Risco diversificável e não diversificável	42
3.2	Correlação	43
3.3	Curvas Risco \times Retorno	45
3.4	Fronteira Eficiente	45
3.5	Princípio da Dominância	46
3.6	Curvas de Utilidade	47
3.7	Capital Market Line	50
3.8	Equação da reta	51
3.9	Carteira Ótima	51
4.1	<i>Ranking</i> Tiobe	53
4.2	Ilustração do Tutorial de acesso ao <i>Colab</i>	54
4.3	Ilustração da página de Documentos do <i>Google Drive</i>	55
4.4	Ilustração do Ambiente <i>Colab</i>	55
4.5	Ilustração de inserção de texto e código do <i>Colab</i>	55
4.6	Ilustração de como renomear título de <i>Notebook</i> do <i>Colab</i>	56
4.7	Ilustração de como copiar <i>Notebook</i> do <i>Colab</i>	56
4.8	Ilustração de como compartilhar <i>Notebook</i> do <i>Colab</i>	57
5.1	Carteira criada em planilha e salva como ativos.csv	69
5.2	Data Frame carregado com a carteira pelo <i>Pandas</i>	70
5.3	Resultado do comando df.describe()	71
5.4	Soma dos investimentos agrupados em categoria.	72
5.5	Arquivo carregado do tesouro direto	74
5.6	Informações do ‘Data Frame’ para o tesouro direto com o comando .info()	79
5.7	Histórico de valores em Real a partir do valor em Dólar	80
5.8	Obtenção da cotação do Real para cada dólar.	82
5.9	‘Data Frame’ final e histórico de cotações da carteira no período de 09/01/2023 a 16/11/2023.	86
5.10	Gráficos de históricos dos ativos.	88

5.11	Histórico de desempenho de valores dos ativos	89
5.12	Retornos de toda carteira e de suas categorias.	90
5.13	Retornos diários de alguns ativos da carteira.	92
5.14	Matriz correlação por mapa de calor nos ativos de renda variável	93
5.15	Risco mínimo resultante dos portfólios aleatórios, retorno esperado e pesos associados, com também a alocação da carteira de renda variável sob tal risco.	97
5.16	Risco mínimo resultante dos portfólios aleatórios, retorno esperado e pesos associados, com também a alocação da carteira de renda variável sob tal risco.	99
5.17	Portfólios aleatórios. Pontos de mínimo risco. Fronteira eficiente.	100
5.18	Resultados de mínimo risco e carteira ótima da renda variável obtidos por otimização.	101
5.19	Carteiras ótimas vindas dos portfólios aleatórios e de otimização.	103

Lista de Quadros

2.1	Competência e habilidades em Estatística - BNCC	21
2.2	Dissertações do Profmat (até a data de 26/09/2023).	22
2.3	Estatística no Modelo de Markowitz (segundo Alves (2020) e Dias (2018)).	23

Sumário

1	Introdução	11
1.1	Justificativa	12
1.2	Motivação da pesquisa	17
1.3	Objetivos	17
1.4	Contribuições do trabalho	18
1.5	Organização do trabalho	18
2	Embasamento Teórico	19
2.1	A estatística e a BNCC	20
2.2	A estatística e os trabalhos do Profmat	22
2.3	Estatística Descritiva	23
2.4	Medidas de tendência central	30
2.5	Medidas de dispersão	32
3	Modelo de formação de carteira	38
3.1	Conceitos iniciais	38
3.2	Modelo de Markowitz	39
3.3	Retorno	40
3.4	Ativo livre de risco	44
3.5	Fronteira Eficiente	44
3.6	Seleção de Carteiras Ótimas	47
4	Python	52
4.1	Noções iniciais	52
4.2	Noções de <i>Python</i>	52
4.3	Primeiros passos	57
4.4	Bibliotecas	65
5	Avaliação de Carteira	67
5.1	Composição da carteira	67
5.2	Importação e tratamentos de dados dos ativos	69
5.3	Retornos dos ativos de renda variável	91
5.4	Geração de portfólios aleatórios de uma carteira	94
5.5	Aproximação para uma carteira ótima.	102
6	Considerações Finais	105

<i>SUMÁRIO</i>	10
Referências	109

1

Introdução

Magalhães & Lima (2023) defendem a estatística como um conhecimento presente em ações cotidianas e exemplifica sua afirmação, lembrando dos dados exibidos durante um evento esportivo. De fato, a análise de dados estatísticos é bastante utilizada, não só nos eventos esportivos, mas nos mais diversos ramos da atuação humana, como é o caso das finanças. Para Dias (2018), a estatística descritiva é de fundamental importância ao gestor de atividades financeiras, por propiciar uma visão sistêmica, auxiliando a tomada de decisão.

Acreditamos que a matemática seja o instrumento capaz de oferecer elementos que facilitem escolhas conscientes, por isso, a temática para intermediar o ensino da matemática e a realidade dos investimentos financeiros, neste trabalho, foi o modelo de Markowitz para a formação de uma carteira de investimentos. Este método, segundo Alves (2020), consiste em um modelo matemático, que através de uma série histórica é capaz de balizar a tomada de decisão, intermediado por elementos pertencentes a estatística.

Para Magalhães & Lima (2023), o principal responsável pela facilitação da obtenção dos dados, bem como sua organização, tratamento e inferência, são os computadores. Pois estes são capazes de abreviar tarefas trabalhosas, através de métodos simples como a aquisição de pacotes estatísticos; ou um pouco mais complexos, como é o caso da linguagem de programação – LP. Certamente, não seria tão fácil, organizar uma matriz, de por exemplo 250 linhas e 20 colunas, tão pouco executar um cálculo envolvendo tantos elementos. Porém, graças ao computador esta tarefa pode ser realizada por um simples comando e entregue em milésimos de segundo.

No ensino básico, segundo Giordano (2017), nos anos 1990 os Parâmetros Curriculares Nacionais – PCN oficializam o ensino da estatística, delegando, em parte, ao professor de matemática a tarefa de ensiná-la. Pois, ainda pela autora, apenas a Universidade Federal do Rio Grande do Sul chegou a ofertar o curso de licenciatura em Estatística por um breve período, deixando de ofertar em 1997.

Em vigor, a Base Nacional Comum Curricular — BNCC, homologada em 14 de dezembro de 2018 para toda a Educação básica, dividiu a Matemática Escolar em compe-

tências e habilidades. Este documento na etapa ensino médio enumera quatro habilidades relacionadas a estatística, que citaremos oportunamente. O referido currículo ainda indica ao ensino médio dar continuidade aos avanços da etapa anterior, cujo foco deve ser o do ensino da matemática aplicada à realidade e com o emprego de tecnologias.

Apesar das inúmeras possibilidades de aplicação, segundo [Alves \(2020\)](#), no ensino básico, o ensino de estatística se encontra desconectado da realidade, frustrando os estudantes ávidos a enxergá-la em seu cotidiano. Ainda segundo o autor, o ensino da estatística segue a fechar os olhos para suas aplicações também no ensino superior, pois nessa fase, geralmente se faz a opção de privilegiar as técnicas de demonstração das fórmulas matemáticas.

A contextualização do ensino da matemática, por meio da temática da educação financeira parece aumentar bastante de importância quando os veículos de imprensa repetem reportagens de títulos similares ao do portal de jornalismo G1: Inadimplência bate novo recorde e atinge 67,6 milhões de brasileiros, aponta Serasa, de 22/08/2022. Corroborando com as preocupações de órgãos internacionais a partir dos anos 2000.

A justificativa deste trabalho será feita, em 3 etapas, nas quais constam: a escolha do objeto do conhecimento, contextualizada por problemas na formação do professor; o tema utilizado para aproximar a estatística a realidade, a educação financeira, escolhido pelo grau de endividamento das famílias indicado por pesquisas oficiais e obrigatoriedade do seu uso, na perspectiva curricular; e o instrumento tecnológico usado para a captura e tratamento de dados com a finalidade de balizar a tomada de decisão.

1.1 Justificativa

1.1.1 A formação de professores e a estatística

[Giordano \(2017\)](#) atenta para o fato da estatística se estabelecer de maneira oficial no currículo da educação básica do Brasil no ano de 1997 para o ensino fundamental e o posterior anúncio em 1999 para o ensino médio, o que nos permite inferir que muitos profissionais que ensinam matemática podem não ter estudado o objeto do conhecimento no ensino básico. Vale a pena ainda ressaltar que mesmo com a inclusão no PCN, não significou a obrigatoriedade do cumprimento por parte dos professores do seu ensino, pois este documento era uma orientação, com o que concluímos que mesmo profissionais formados após este período podem não ter estudado a estatística na referida etapa de ensino.

Os problemas na formação dos professores de matemática que ensinam estatística se tornam evidentes, quando na contramão do que era indicado pelo PCN, segundo [Baccar, Velasque & Leal \(2023\)](#), o Conselho Nacional de Educação – CNE, por intermédio do parecer

nº 1.302/2001, desobrigam os cursos de licenciatura em matemática em ofertar a disciplina estatística.

Já na BNCC, o currículo ganha um caráter obrigatório e [Giordano, Alves & Queiroz \(2019\)](#) identificaram e analisaram as habilidades do grupo de objetos do conhecimento de análise combinatória, probabilidade e estatística que seriam as inclusões mais recentes ao currículo. Tais habilidades estariam presentes em todas as etapas do ensino básico, incluindo o ensino infantil, portanto ampliando o grupo de professores que ensinam estatística no ensino básico para os pedagogos, que não estão contemplados neste trabalho. Porém não descartamos um estudo futuro sobre este tema.

Nos anos finais do ensino fundamental e no ensino médio, os professores que ficam encarregados de ensinar estatística são os licenciados em matemática e analisando esta graduação, [Giordano \(2017\)](#) conclui que a maioria possui uma ou duas disciplinas de estatística: quando possuem uma disciplina, esta está voltada para a probabilidade; já quando duas, a ênfase está na estatística inferencial e em ambos os casos o foco é o ensino superior.

[Baccar, Velasque & Leal \(2023\)](#), apresentaram os resultados de uma pesquisa, que buscava identificar que saberes estatísticos os licenciandos sinalizam trazer da educação básica, bem como o que significa para eles o trabalho com Estatística. Cujos resultados indicaram que, apesar de a maioria dos estudantes sinalizar já ter tido algum tipo de contato com o tema na educação básica e reconhecer a importância da Estatística para a vida atual, este conhecimento era limitado apenas a aspectos procedimentais da estatística descritiva, restringindo-se a alguns gráficos e a cálculos de algumas medidas de centralidade. Além disso, os licenciandos apresentaram dificuldade para falar sobre variabilidade, aleatoriedade e incerteza. E, embora sinalizassem ter conhecimentos sobre probabilidade, afirmaram desconhecer totalmente o que englobava o trabalho com a inferência estatística.

Os estudos supracitados evidenciam as dificuldades na formação de professores, o que demonstra a necessidade de trabalhos que vislumbrem a transposição das dificuldades destes temas. Para nossa abordagem cremos que o ensino deste objeto de conhecimento se torna mais atrativo quando associado a temas relevantes, como é o caso da educação financeira.

1.1.2 A educação financeira

Organismos internacionais como a Organização para Cooperação do Desenvolvimento Econômico — OCDE e o Banco Mundial, provocaram os governos no sentido da implementação da Educação Financeira no Ensino Básico. Segundo [Franzoni & Quartieri \(2020\)](#), a justificativa se dá pela necessidade de escolhas, cada vez mais precoce, de produtos financeiros adequados aos objetivos pessoais; o grau de endividamento das famílias; e o consumismo.

Para [Ribeiro \(2020\)](#) o tema da inclusão financeira como combate à pobreza ganha

maior relevância a partir dos anos 2000, tanto que a OCDE, em 2003 já fomentava a Educação Financeira como instrumento para tal inclusão. Segundo a autora, no Brasil, o tema começa a ser discutido em 2007 e em 2010 se torna política pública com a implementação da Estratégia Nacional de Educação Financeira – ENEF por meio do decreto presidencial 7397/2010 e renovado pelo Decreto Federal nº 10.393, de 9 de junho de 2020, cujos objetivos eram:

Promover e fomentar uma cultura de Educação Financeira no país; ampliar a compreensão dos cidadãos para que possam fazer escolhas bem informadas sobre a gestão de seus recursos; contribuir para a eficiência e solidez dos mercados financeiro, de capitais, de seguros e de fundos de previdência (BRASIL, 2010, p. 20)

Para Kuntz (2019) este documento reforça a intenção de fomentar a educação financeira para o maior número de pessoas possíveis, por acreditar que dotados desses conhecimentos os cidadãos tomem decisões em suas finanças pessoais mais assertivas, de modo a facilitar a realização de objetivos pessoais.

O conceito de Educação Financeira utilizada pelo ENEF (2010), está consoante com o conceito da OCDE (2005) encontrado no seu plano diretor e em diversos trabalhos sobre o tema, como os de (SODRÉ, 2018, p. 53); e (SILVA, 2021, p. 24):

“o processo mediante o qual os indivíduos e as sociedades melhoram sua compreensão dos conceitos e dos produtos financeiros, de maneira que, com informação, formação e orientação claras, adquiram os valores e as competências necessários para se tornarem conscientes das oportunidades e dos riscos neles envolvidos e, então, façam escolhas bem informados, saibam onde procurar ajuda, adotem outras ações que melhorem o seu bem-estar, contribuindo, assim, de modo consistente para formação de indivíduos e sociedades responsáveis, comprometidos com o futuro”(BRASIL, 2010, p. 20).

1.1.2.1 As pesquisas sobre a realidade financeira dos brasileiros

A falta de habilidade dos brasileiros em gerir seus recursos financeiros, vem sendo retratada em pesquisas oficiais e seus números têm justificado inúmeros trabalhos, como os de Sodré (2018); Campos (2021); e Pessoa (2022).

Segundo Sodré (2018), em seu plano diretor a ENEF (2010) usou dados da Pesquisa Nacional Financeira, que foi promovida com parceria com a BM&FBOVESPA, que exibiu como resultado um baixo conhecimento do brasileiro na gestão de seus recursos pessoais, falta de planejamento financeiro e a vulnerabilidade a golpes. Tais resultados justificaram a implementação da ENEF. A referida pesquisa teve os seguintes temas: “planejamento financeiro, economia, serviços financeiros, crédito e juros, investimentos, previdência social, seguros, capitalização, e proteção e defesa do consumidor” (BRASIL, 2010, p. 4).

Segundo Campos (2021), a Pesquisa de Endividamento e Inadimplência do Consumidor - PEIC, que em outubro de 2021 alertou para o grau de endividamento das famílias brasileiras, evidenciou através de comparativos que famílias com menor poder aquisitivo eram mais endividadas; e apontou que a Educação Financeira escolar poderia contribuir com uma mudança para um quadro que privilegie a cultura da poupança e planejamento, capaz de reverter a realidade dos brasileiros.

Pessoa (2022) usou a pesquisa do Serviço de Proteção ao Crédito - SPC realizada no ano de 2019 para concluir que parcela considerável da população brasileira tem dificuldade na gerência de seus recursos e aponta como causa a desarticulação entre a realidade e o ensino da matemática.

Em conformidade com as pesquisas citadas, em 2022, o Serviço de Assessoria – SERASA divulga um estudo em que constata nove meses seguidos do crescimento da inadimplência no Brasil e o perfil dos pesquisados foi o de: brancos - 33%; solteiros – 44% e de idade entre os 18 aos 24 anos – 20%. A principal causa do endividamento é o desemprego – 29% e o cartão de crédito é a principal dívida para 53% dos entrevistados. Ainda segundo esta pesquisa, os endividados enfrentam problemas emocionais como: crises ou momentos de ansiedade – 61%; sentiram muita tristeza e medo do futuro – 53%; e passaram a não confiar na sua capacidade de tomar suas decisões e cuidar de suas finanças.

Todos esses estudos corroboram as preocupações já mencionadas, em Franzoni & Quartieri (2020); e Ribeiro (2020), de organismos internacionais e creem na educação escolar como fator para mudança da realidade financeira, intermediada por informações que permitam ao cidadão a melhor decisão de seus produtos financeiros.

1.1.3 Currículo e a Educação Financeira

Para o Ensino Fundamental, Brasil (2018) organiza as habilidades da área de Matemática e suas Tecnologias em 5 unidades, a saber: Números, Álgebra, Geometria, Grandezas e medidas e Probabilidade e Estatística. Ainda nesta etapa, é na Educação Financeira, onde são “discutidos assuntos como taxas de juros, inflação, aplicações financeiras (rentabilidade e liquidez de um investimento) e impostos” (BRASIL, 2018, p. 269). Determina a abordagem da Educação Financeira por meio de projetos interdisciplinares, assim como temas com os quais a Matemática Financeira poderia contribuir. Como os que já eram recomendados nos Parâmetros Curriculares Nacionais da Educação Fundamental – PCNEF(1998):

Para compreender, avaliar e decidir sobre algumas situações da vida cotidiana, como qual a melhor forma de pagar uma compra, de escolher um financiamento etc., é necessário trabalhar situações-problema sobre a Matemática Comercial e Financeira, como calcular juros simples e compostos e dividir em partes proporcionais pois os conteúdos necessários para resolver

essas situações já estão incorporados nos blocos (BRASIL, 1998, p. 86).

Na etapa Ensino Médio, a Educação Financeira também é uma temática a ser abordada de maneira interdisciplinar, nas Ciências Humanas e Sociais Aplicadas, ressalta as práticas empreendedoras e “a importância da educação financeira e da compreensão do sistema monetário contemporâneo nacional e mundial, imprescindíveis para uma inserção crítica e consciente no mundo atual” (BRASIL, 2018, p.568). Ideia consoante as encontradas nos Parâmetros Curriculares Nacionais do Ensino Médio - PCNEM (2000):

Em um mundo onde as necessidades sociais, culturais e profissionais ganham novos contornos, todas as áreas requerem alguma competência em Matemática e a possibilidade de compreender conceitos e procedimentos matemáticos é necessária tanto para tirar conclusões e fazer argumentações, quanto para o cidadão agir como consumidor prudente ou tomar decisões em sua vida pessoal e profissional (BRASIL, 2000, p. 40).

Franzoni & Quartieri (2020); e Ribeiro (2020) apontam para a grande relevância da Educação Financeira para a sociedade, por isso defendem a inclusão da temática na escola, visto que as pesquisas, como: SPC (2019); PEIC (2021); e SERASA(2022) revelam um quadro preocupante da gestão dos recursos pessoais, em que a Educação Financeira escolar poderia contribuir para a mudança desta realidade. Brasil (2018) aponta para a adoção do tema por meio de projetos interdisciplinares, ou seja, envolvendo outras disciplinas que não só a matemática.

Este trabalho, alinhado com o que indicavam os organismos internacionais, com a BNCC, e com trabalhos científicos justificados pela situação de endividamento dos brasileiros, procurou fomentar o ensino de estatística pelo relevante tema da educação financeira e usando de instrumentos tecnológicos, como prevê nosso currículo.

1.1.4 O currículo e o uso das tecnologias

Desde o *An agenda for action: recommendations for School Mathematics of 1980s*, que foi um documento produzido pelo *National Council of Teachers of Mathematics – NCTM*, onde constam oito recomendações para a reestruturação e a organização curricular da Matemática escolar básica da época, entre elas o de obter vantagem do uso de tecnologia, como: computadores e celulares, que os currículos vêm apontando para o uso destas, inclusive “a BNCC propõe que os estudantes utilizem tecnologias, como calculadoras e planilhas eletrônicas, desde os anos iniciais do Ensino Fundamental” (BRASIL, 2018, p. 528).

Para o nosso trabalho, usamos o computador, mais especificamente o uso da LP *python*, intermediada pela *Integrated Development Environment - IDE¹ google colab*, que

¹Ambiente integrado de desenvolvimento, tradução nossa

é uma versão online da LP. Na BNCC, encontramos apenas uma competência, qual seja: “(EM13MAT405) Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática” (BRASIL, 2018, p. 539).

Por fim, justificamos este trabalho por oferecer um material para o ensino da estatística na etapa ensino médio, onde os profissionais que ensinam estatística, nesta etapa, segundo Giordano (2017); Giordano, Alves & Queiroz (2019); Baccar, Velasque & Leal (2023) apresentam dificuldades formativas. Para aproximar o objeto do conhecimento, escolhemos a temática da educação financeira, conforme indicações de organismos internacionais; Brasil (2018); e trabalhos científicos de Sodr e (2018); Campos (2021) e Pessoa (2022) que usaram como justificativa pesquisas oficiais que retratam a realidade da m  gest o dos recursos financeiros pelos brasileiros. E por  ltimo, o uso da tecnologia, conforme preconiza Brasil (2018).

1.2 Motivação da pesquisa

Ao usar assuntos de investimentos aplicando o modelo de Markowitz para a formaç o de carteira, bem como a otimizaç o destas e ferramentas computacionais do *google colab*, este trabalho   motivado pelas possibilidades que se abrem quando se faz a conex o de um assunto te rico (como   o caso da estat stica) com a realidade do dia a dia (Educaç o Financeira).

Considerando a motivaç o, estabelecemos os seguintes objetivos:

1.3 Objetivos

O objetivo geral deste trabalho foi o de fomentar o ensino da estat stica, na tem tica da educaç o financeira e intermediado por tecnologia. Para atingir a este objetivo, almejamos os seguintes objetivos espec ficos:

- Indicar, por meio de pesquisa bibliogr fica em trabalhos e documentos oficiais, a delimitaç o do objeto do conhecimento, a Estat stica;
- conhecer o m todo para formaç o de carteira de Markowitz, bem como o da otimizaç o da carteira;
- apresentar o *google colab*, assim como alguns comandos; e
- aplicar o m todo para uma carteira de investimentos.

1.4 Contribuições do trabalho

Esperamos que este trabalho contribua, no sentido de: fomentar o ensino da estatística por meio da temática da educação financeira; usar a Linguagem de programação *Python*, em sua versão online *google colab*, com a intenção do ensino da estatística; e contribuir com futuras pesquisas acadêmicas que envolvam os temas supramencionados.

1.5 Organização do trabalho

Este trabalho obedece a seguinte organização: os capítulos 2, 3 e 4, fazem parte do embasamento teórico; o capítulo 5 é o desenvolvimento do trabalho ou resultados da pesquisa; e logo após este, teremos as considerações finais. Cujos resumos seguem:

No Capítulo 2 realizamos uma pesquisa bibliográfica, que delimitou o objeto do conhecimento a ser estudado, a estatística, sobretudo a descritiva, que cremos ser compatível com a realidade dos estudantes do ensino médio.

No Capítulo 3 identificamos a realidade que resolvemos contextualizar este objeto: a Educação Financeira, mais especificamente: o Modelo de Markowitz para a formação de uma carteira de investimentos, bem como sua otimização.

No Capítulo 4 apresentamos um breve tutorial da IDE *google colab* da LP *python*, que foi responsável pela obtenção, organização e tratamento dos dados, bem como um apinhado de comandos que foram utilizados no capítulo posterior.

No Capítulo 5 utilizaremos o modelo estatístico proposto por Markowitz, bem como a solução da carteira ótima para um conjunto de ativos, intermediado pelo *python*.

Por fim faremos nossas considerações finais, onde revisitaremos nosso trabalho com a finalidade de concluir o cumprimento do nosso objetivo de fomentar o ensino de estatística usando o Modelo de Markowitz: intermediado pelo *python*.

Embasamento Teórico

Neste capítulo registramos resultados de uma pesquisa sobre a Base Nacional Comum Curricular – BNCC e na base de dissertações do Profmat, que nos permitiram uma melhor delimitação do objeto do conhecimento matemático, no caso a estatística, em torno do modelo de Markowitz para a formação de carteiras.

Segundo [Giordano \(2017\)](#) a palavra Estatística é proveniente do latim *statisticum* que significa relativo ao estado. Teve sua origem ligada ao curso de Ciências Sociais e Políticas, com características censitárias, cujo objetivo era o de auxiliar no gerenciamento do estado, instrumentalizando a tomada de decisão de governantes. Ainda pela mesma autora, nos séculos XIX e XX, a estatística passa a ser usada como método científico e por este motivo, vira disciplina em cursos superiores e técnicos nas diversas áreas do conhecimento. No ensino básico, seu ensino coube aos pedagogos no ensino infantil e primeiros anos do ensino fundamental, enquanto que nos últimos anos do ensino fundamental e ensino médio ao professor de matemática.

A estatística para [Iezzi, Hazzan & Degenszajn \(2013\)](#) se ocupa da coleta, apresentação e interpretação de um determinado conjunto de dados. Os dados são segundo [Magalhães & Lima \(2023\)](#), obtidos de inúmeras áreas de atuação humana, podendo estes serem ou não numéricos.

[Giordano \(2017\)](#) divide a estatística em descritiva e inferencial:

A Estatística descritiva visa retratar a realidade ou a experiência em estudo, a fim de compreendê-la localmente, no seu contexto. Para isso utiliza métodos de coleta, tabulação, resumo e representação de dados na intenção de destacar alguns aspectos essenciais, tais como: padrões, regularidades e tendências do conjunto observado. São elementos de estudo: tabelas, gráficos, medidas de posição (média, moda, mediana, percentis, quartis) e de dispersão (amplitude, variância, desvio padrão).

A Estatística inferencial, por sua vez, objetiva efetuar generalizações baseadas em amostras estatísticas de populações que não podem ser estudadas na sua totalidade. Dessa forma, debruça-se sobre estudos probabilísticos na busca de modelos matemáticos que possibilitem a extrapolação

das conclusões alcançadas localmente para uma população maior. Os testes de hipóteses são exemplos de objeto de estudo da Estatística inferencial. (GIORDANO, 2017, p. 31)

Ou seja, na estatística descritiva o foco é o de coleta e organização de dados, enquanto a estatística inferencial trata de interpretá-los e fazer previsões, que não obrigatoriamente serão confirmadas. Como é o caso de estudos com a temática do modelo de Markowitz para a formação de carteira, pois este se baseia em uma série histórica, ou seja, onde os dados coletados pertencem ao passado, com o objetivo de “projetar” o futuro ou servir como balizador para previsões e tomada de decisões de ativos na bolsa de valores, enfatizando que estas podem ou não ocorrer.

2.1 A estatística e a BNCC

A BNCC, é um documento curricular com força de lei que normatiza as aprendizagens essenciais que os estudantes precisam desenvolver para que tenham os direitos à aprendizagem e desenvolvimento garantidos; além de nortear as reformulações curriculares de Estados e Municípios, assim como o Projeto Político Pedagógico (PPP) das escolas.

A proposta do ensino da matemática, para o ensino médio, segundo Brasil (2018), é o da ampliação de conhecimentos oriundos do ensino fundamental. Em relação à estatística, está presente em todas as etapas do ensino básico. E mais,

Para o desenvolvimento de habilidades relativas à Estatística, os estudantes têm oportunidades não apenas de interpretar estatísticas divulgadas pela mídia, mas, sobretudo, de planejar e executar pesquisa amostral, interpretando as medidas de tendência central, e de comunicar os resultados obtidos por meio de relatórios, incluindo representações gráficas adequadas. (BRASIL, 2018, p. 518)

Em relação as competências e habilidades listadas na BNCC do ensino médio, em estatística, temos:

Quadro 2.1: Competência e habilidades em Estatística - BNCC

Competências	Habilidades
Utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam atividades cotidianas, sejam fatos das Ciências da Natureza e Humanas, ou ainda questões econômicas ou tecnológicas, divulgados por diferentes meios, de modo a consolidar uma formação científica geral.	(EM13MAT102) Analisar gráficos e métodos de amostragem de pesquisas estatísticas apresentadas em relatórios divulgados por diferentes meios de comunicação, identificando, quando for o caso, inadequações que possam induzir a erros de interpretação, como escalas e amostras não apropriadas.
Articular conhecimentos matemáticos ao propor e/ou participar de ações para investigar desafios do mundo contemporâneo e tomar decisões éticas e socialmente responsáveis, com base na análise de problemas de urgência social, como os voltados a situações de saúde, sustentabilidade, das implicações da tecnologia no mundo do trabalho, entre outros, recorrendo a conceitos, procedimentos e linguagens próprios da Matemática.	(EM13MAT202) Planejar e executar pesquisa amostral usando dados coletados ou de diferentes fontes sobre questões relevantes atuais, incluindo ou não, apoio de recursos tecnológicos, e comunicar os resultados por meio de relatório contendo gráficos e interpretação das medidas de tendência central e das de dispersão.
Utilizar estratégias, conceitos e procedimentos matemáticos, em seus campos – Aritmética, Álgebra, Grandezas e Medidas, Geometria, Probabilidade e Estatística – para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente.	(EM13MAT316) Resolver e elaborar problemas, em diferentes contextos, que envolvem cálculo e interpretação das medidas de tendência central (média, moda, mediana) e das de dispersão (amplitude, variância e desvio padrão).
Compreender e utilizar, com flexibilidade e fluidez, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas, de modo a favorecer a construção e o desenvolvimento do raciocínio matemático.	(EM13MAT408) Construir e interpretar tabelas e gráficos de frequências, com base em dados obtidos em pesquisas por amostras estatísticas, incluindo ou não o uso de softwares que interrelacionem estatística, geometria e álgebra

FONTE: Brasil (2018, p. 532 - 539)

Realizando uma análise das competências e habilidades podemos concluir que o ensino/aprendizagem deste objeto do conhecimento, deve se ocupar em: Construção, leitura

e interpretação de gráficos e tabelas; Medidas de tendência central: Média; Moda; e Mediana, e Medidas de dispersão, quais sejam: Amplitude, Variância e Desvio Padrão. Com ou sem o uso de *software*.

2.2 A estatística e os trabalhos do Profmat

Ao consultarmos o banco de dissertações do programa do Mestrado Profissional em Matemática em Rede Nacional – Profmat, digitamos *Markowitz* no campo destinado para título da dissertação, no dia 26/09/2023 às 23:32, e encontramos as seguintes dissertações:

Quadro 2.2: Dissertações do Profmat (até a data de 26/09/2023).

Data de defesa	Aluno	Título da Dissertação	Instituição
17/06/2020	Delmário Prates Alves	A Estatística como Auxílio na Seleção de Carteiras de Investimento – Uma Aplicação por meio do Modelo de Markowitz	UNB
14/11/2018	Denis Pereira Dias	Teoria de Markowitz e Programação Linear para Formação de uma Carteira Ótima de Investimentos	UFSCAR

FONTE: Dissertações do PROFMAT (Dissertações com título a palavra *Markowitz*).

Alves (2020) apresentou como objetivo geral: mostrar a importância da Estatística na decisão de investimento, por meio da aplicação do modelo proposto por Markowitz, apresentando-o como uma opção de ferramenta de apoio ao investidor iniciante na seleção de uma carteira de ativos. Para cumprir este objetivo, usou como metodologia a pesquisa bibliográfica, cujos resultados, foram os aportes teóricos do modelo de Markowitz e os conceitos estatísticos que o envolvem. Ao final do trabalho, discorreu sobre como a estatística pode ser ensinada aos estudantes do ensino médio, de modo a conectá-la a realidade.

Enquanto Dias (2018) apresentou como objetivo a seleção de uma carteira ótima de investimentos. Para atingir este objetivo usou revisão de literatura, que resultou na escolha da Programação Linear atrelada à Teoria Moderna de Portfólios de Markowitz para modelar e resolver dois problemas de programação linear; que resultaram na composição de duas carteiras considerando os perfis de investidores, conservador e moderado. Para a modelagem dos problemas foi realizada uma análise técnica acerca dos ativos que foram selecionados previamente para a composição da carteira. A seleção dos ativos levou em consideração alguns parâmetros estatísticos, tais como média aritmética ponderada e coeficiente de correlação. Os resultados obtidos corroboraram com as previsões de alguns especialistas, que afirmavam

que o ano de 2018 seria favorável à investimentos na bolsa de valores. A conclusão foi que, os resultados não são uma garantia da eficiência na prática, porém pode ser um bom balizador para futuras atividades de investimento, ou, de análises/associação de eventos ocorridos no período aos rendimentos de diversos tipos de investimentos.

Segundo os autores, são elementos da estatística, atrelados ao modelo de Markowitz para a formação de carteira:

Quadro 2.3: Estatística no Modelo de Markowitz (segundo [Alves \(2020\)](#) e [Dias \(2018\)](#)).

Autor	Tópicos da estatística
Alves (2020)	Variáveis Aleatórias; Valor Esperado ou Esperança Matemática; Variância e Covariância; Coeficiente de Correlação; Coeficiente de Variação; Distribuição Normal e suas Implicações para o Desvio Padrão.
Dias (2018)	Medidas de Centralidade e Variabilidade; Média Aritmética; Média Aritmética Ponderada; Variância; Desvio Padrão; Escore Padronizado; Coeficiente de Variação; Covariância; Coeficiente de Correlação.

FONTE: [Alves \(2020\)](#) e [Dias \(2018\)](#).

Analisando os estudos citados, quais sejam: [Brasil \(2018\)](#); [Alves \(2020\)](#); e [Dias \(2018\)](#); podemos concluir que existem muitos pontos de convergência entre esses documentos e estes receberam especial atenção neste trabalho. Alguns itens, muito embora extrapolem o currículo para o ensino médio, não cremos ser de impossível compreensão para este público. Baseado nessa crença, passemos a buscar os conceitos e definições do nosso objeto de estudo, a Estatística.

2.3 Estatística Descritiva

2.3.1 População e Amostra

Estatisticamente, segundo [Magalhães & Lima \(2023\)](#), população é o grande conjunto de dados, cujas características são objetos de interesse, já a amostra é um subconjunto notoriamente menor da população. Ainda, segundo os autores, as pesquisas do tipo amostral devem ser realizadas quando não for possível colher os dados da população, por motivos monetários ou geográficos por exemplo.

2.3.2 Variáveis

Para [Iezzi, Hazzan & Degenszajn \(2013\)](#), as variáveis estatísticas visam responder a um determinado questionamento. Quando a resposta deste questionamento é um substantivo, dizemos que a variável é qualitativa. Agora, se a resposta for representada por um número, dizemos que a variável é quantitativa. [Magalhães & Lima \(2023\)](#) subdividem as variáveis qualitativas em: Nominal quando não se pode estabelecer uma ordem entre seus valores, e Ordinal quando é possível estabelecer uma ordem natural. As quantitativas em: Discreta, quando se assume valores pertencentes ao conjunto dos números inteiros; e Contínua, quando seus valores são expressos por valores pertencentes ao conjunto/intervalo dos números reais.

Um outro tipo de variável cuja definição é muito útil ao nosso trabalho, é o de variável aleatória, que encontramos em [Alves \(2020\)](#):

Variável aleatória: Seja ω um experimento aleatório¹ e Ω o espaço amostral² associado ao experimento. Uma função X que associe a cada elemento $\omega \in \Omega$ um número real $X(\omega)$ é denominada variável aleatória (v.a.).

2.3.3 Organização dos dados

Por vezes a leitura dos dados brutos, maneira pela qual são obtidos, dificultam a leitura e a visualização dos dados. Para melhorar esse acesso à informação existem algumas formas de organizá-los, entre elas, em tabela de frequência e gráficos.

2.3.3.1 Tabela Frequência

De modo mais geral podemos definir uma tabela como a organização dos dados em linhas e colunas. Geralmente, as tabelas de frequência possuem mais de uma coluna; em uma temos os resultados ou expectativas de resultados para um determinado questionamento ao qual temos interesse em saber o resultado, e em outras podem constar as frequências: absoluta – F_A ; total – F_T ; e/ou relativa – F_i ; que [Iezzi, Hazzan & Degenszajn \(2013\)](#) tem referenciado por:

- Frequência Absoluta – F_A : descreve a quantidade de vezes que um determinado valor se repete num conjunto de dados de n elementos, que descreveremos por n_i ;

¹situação ou experimento que não podem ser previstos com certeza ([MAGALHÃES; LIMA, 2023](#), p. 49)

²conjunto de todos os resultados possíveis de um certo fenômeno aleatório ([MAGALHÃES; LIMA, 2023](#), p. 49)

- Frequência total – F_T : é o somatório das frequências absolutas, ou seja,

$$F_T = \sum_{i=1}^n n_i; \text{ e} \quad (2.1)$$

- Frequência relativa – F_i : é a razão entre cada uma das frequências absolutas e frequência total; ou seja,

$$F_i = \frac{n_i}{n} \quad (2.2)$$

Ainda segundo o autor, a frequência relativa pode ser representada por uma fração, por um número decimal, ou em forma de porcentagem. O intervalo da frequência relativa é $0 \leq F_i \leq 1$, pois $0 \leq n_i \leq n$. Se a representação da F_i estiver em forma fracionária ou decimal o somatório das frequências relativas é igual a 1; e se em forma de porcentagem, a somatória será 100%.

Um exemplo simples seria o de uma tabela representando o lançamento de um dado simples, com 10 repetições. Para as frequências faremos uma tabela de 3 colunas, onde: na primeira escrevemos os resultados possíveis para o lançamento de um dado simples; na segunda os resultados das F_A para cada resultado, e no final da segunda coluna adicionaremos uma célula com a somatória das F_A , ou seja F_T ; e na terceira coluna escreveremos as F_i na representação decimal; e no final desta coluna adicionaremos a somatória dos F_i , que iremos representar por SR , esperando encontrar o valor igual a 1.

Tabela 2.1: Tabela de frequência de uma certa ocorrência de valores em 10 lançamentos de um dado.

Resultados esperados	F_A	F_i
1	4	0,4
2	2	0,2
3	0	0
4	1	0,1
5	0	0
6	3	0,3
	$F_T = 10$	$SR = 1$

Fonte: do autor.

Temos que os valores esperados para o lançamento de um dado são 1, 2, 3, 4, 5, 6, e pela tabela o valor 1 ocorreu em 4 oportunidades, logo $F_1 = 0,4$, resultado da razão entre $n_1 = 4$, e $F_T = 10$. Analogamente, os resultados 3 e 5 não ocorreram, por isso $F_3 = F_5 = 0$; o resultado 2 ocorreu duas vezes; o resultado 4 uma vez; e 6 ocorreram 3 vezes, daí $F_2 = 0,2$; $F_4 = 0,1$ e $F_6 = 0,3$, respectivamente.

Um exemplo mais voltado para o tema do nosso trabalho, usando a tabela para organizar os dados em linhas e colunas, seria o de coletar os dados de uma ação do mercado varejista, o Magazine Luiza no caso, numa sequência de 5 dias consecutivos:

Figura 2.1: Informações do ativo

```

[*****100%*****] 1 of 1 completed
      Open  High  Low  Close  Adj Close  Volume
Date
2023-09-18  2.48  2.62  2.45  2.58      2.58  109649300
2023-09-19  2.58  2.61  2.50  2.52      2.52   64918600
2023-09-20  2.52  2.57  2.51  2.52      2.52   57258300
2023-09-21  2.47  2.48  2.35  2.35      2.35  135662700
2023-09-22  2.39  2.41  2.17  2.24      2.24  133978200

```

Fonte: do autor.

Usando a versão online do *Python*, no caso o *colab notebook*, com alguns comandos, que descreveremos mais tarde, obtivemos os dados da bolsa de valores de 5 dias de funcionamento da semana de segunda-feira, 18/09/2023 à sexta-feira 22/09/2023. Estes dados estão dispostos em seis linhas e 7 colunas. Na primeira coluna, temos a data a que se refere os dados; na segunda o preço na abertura dos trabalhos; na terceira o maior preço; na quarta o menor preço; na quinta o preço do fechamento; na sexta o preço do fechamento ajustado; e a sétima o volume de negociações dessa ação no dia. Ao observar a tabela, seria fácil notar, por exemplo, que após o fechamento ajustado no dia 19/09/2023, o preço do ativo foi de R\$ 2,52 e o volume negociado foi de 64.918.600 reais.

Outra maneira, bastante comum de organização de dados é por meio de um gráfico. Gráfico tem apelo visual, por isso é muito comum o uso em revistas, jornais impressos e televisivos, apresentações acadêmicas entre outras.

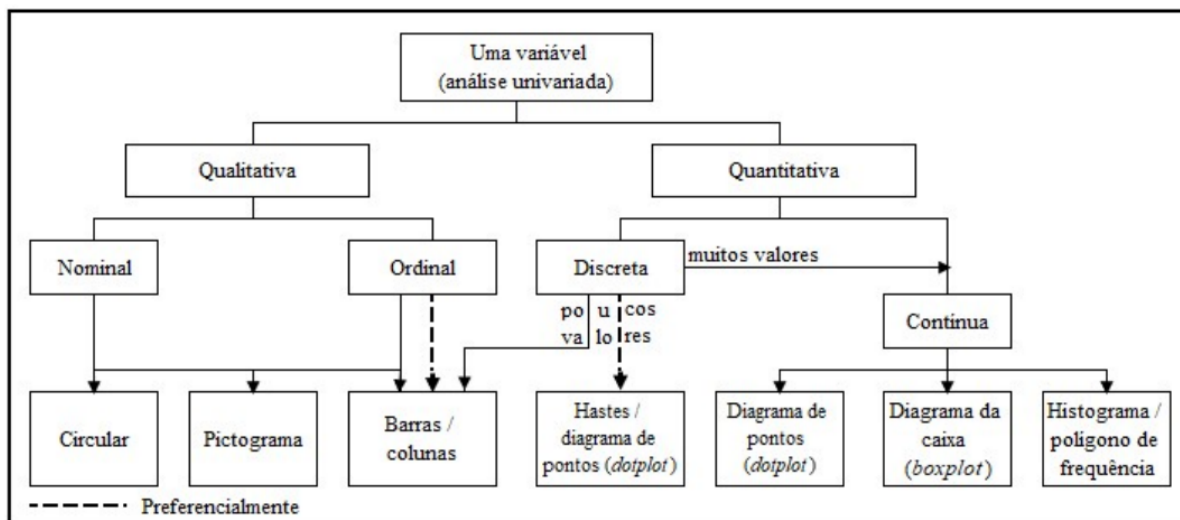
2.3.4 Gráficos

Para Cazorla, Utsumi & Monteiro (2020), o tipo de gráfico a ser escolhido, depende do tipo de variável que está sendo estudada. Eles fazem indicações conforme se ilustra a Figura 2.2:

Gráfico Circular

É também conhecido como de setores, por ser formado por setores circulares, onde cada setor corresponde a frequência que um elemento ocupa no conjunto de dados. Corroborando com Cazorla, Utsumi & Monteiro (2020); e Magalhães & Lima (2023) também o recomenda para

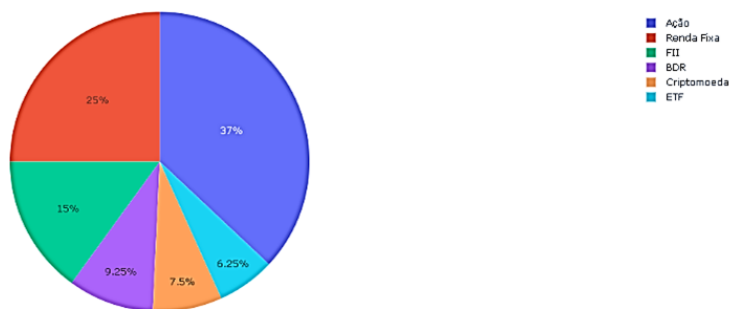
Figura 2.2: Tipos de Gráfico



Fonte: Cazorla, Utsumi & Monteiro (2020, p. 192)

variáveis do tipo qualitativa. Como exemplo, o gráfico da Figura 2.3 mostra a porcentagem que determinados tipos de ativo em uma carteira de investimentos.

Figura 2.3: Gráfico circular



Fonte: do autor.

Observamos que para essa representação gráfica foi usado uma legenda de cores, o que facilita sua leitura. Por exemplo podemos perceber visualmente que: o ativo ação corresponde a 37%; Renda fixa a 25%; FII³ 15%; BDR⁴ 9,25%; Criptomoedas 7,5%; ETF⁵ 6,5%.

Uma outra forma de praticar as habilidades, seria o de a partir de uma tabela

³Fundos de investimento imobiliário: ativos ofertados na bolsa, para direito de recebimento em parte de aluguéis cobrados em empreendimentos comerciais e residenciais, entre outros;

⁴Brazilian Depositary Receipt: são certificados de ações de empresas internacionais, listadas por exemplo nas bolsas americanas; e

⁵sigla em inglês para fundos de investimentos negociados em bolsa de valores. São fundos listados na bolsa de valores. Tais fundos são comumente procurados por quem objetiva diversificação.

construir o gráfico de setores. Para isso, retomemos a tabela do exemplo de lançamento 10 vezes do dado, e acrescentaremos uma coluna, a frequência relativa em porcentagem, que é resultante da multiplicação da coluna F_i por 100. Como percebemos na Tabela 2.2.

Tabela 2.2: Tabela Frequência

Resultados esperados	F_A	F_i	
1	4	0,4	40%
2	2	0,2	20%
3	0	0	0%
4	1	0,1	10%
5	0	0	0%
6	3	0,3	30%
-	$F_A = 10$	$SR = 1$	$SR = 100\%$

Fonte: do autor.

Para calcularmos o ângulo que ocupará cada frequência relativa ao resultado - α_1 no setor circular, basta fazer:

$$\alpha_1 = \frac{360^\circ \cdot 40\%}{100\%} = 144^\circ.$$

Analogamente para os resultados: 2, 4 e 6. Obtendo:

$$\alpha_2 = \frac{360^\circ \cdot 20\%}{100\%} = 72^\circ.$$

$$\alpha_4 = \frac{360^\circ \cdot 10\%}{100\%} = 36^\circ.$$

$$\alpha_6 = \frac{360^\circ \cdot 30\%}{100\%} = 108^\circ.$$

Para os resultados 3 e 5, terão o mesmo ângulo, pois possuem a mesma $F_A = 0$. Temos:

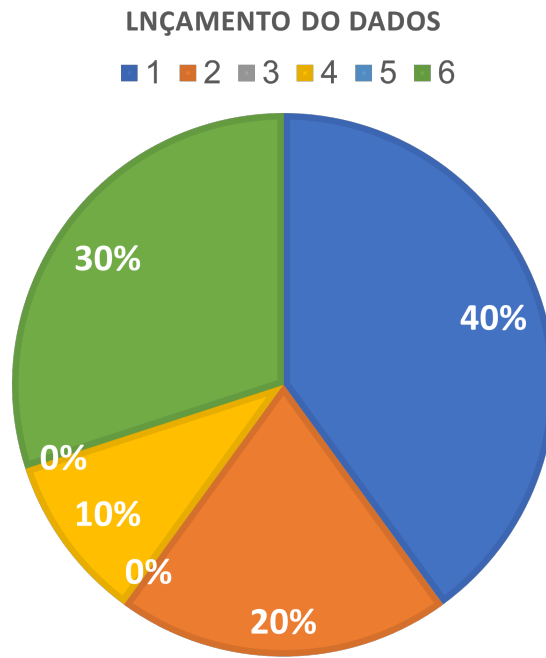
$$\alpha_3 = \alpha_5 = \frac{360^\circ \cdot 0\%}{100\%} = 0^\circ.$$

O gráfico está representado na Figura 2.4

Gráfico de linhas

“O gráfico de linhas é muito usado quando se quer representar o comportamento de uma variável cujos valores diminuem ou aumentam no decorrer do tempo de maneira contínua”. (MAGALHÃES; LIMA, 2023, p. 102). Dependendo da variável que se apresenta nele, pouca informação é retornada. Cabe ao interessado manipular os dados com a variável a fim de extrair significação. Por exemplo, o gráfico de linha da Figura 2.5 abaixo refere-se aos valores

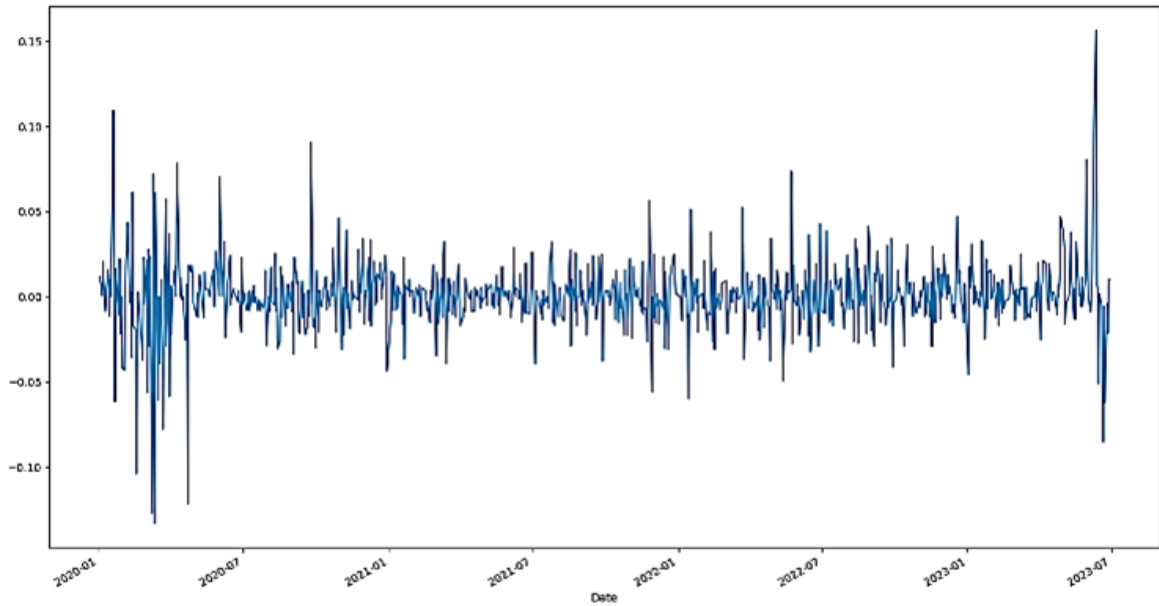
Figura 2.4: Representação dos ângulos do setor circular



Fonte: do autor

da razão do preço de um ativo com o seu valor no dia anterior, menos 1. Ele mostra apenas que o ativo tem uma grande variação desse cálculo no decorrer do tempo!

Figura 2.5: Gráfico de linhas



Fonte: do autor

2.4 Medidas de tendência central

Segundo Dias (2018), as medidas de centralidade são aquelas que visam representar um conjunto de dados por um único número, são medidas de tendência central: média aritmética; média aritmética ponderada; moda e mediana. As definições contidas sobre tais objetos podem ser encontradas em Iezzi, Hazzan & Degenszajn (2013); Bussab & Morettin (2017); e Magalhães & Lima (2023).

2.4.1 Média aritmética

Seja x uma variável quantitativa e x_1, x_2, \dots, x_n os valores assumidos por x . Define-se a média aritmética de x – indicada por x_m – como a divisão da soma de todos esses valores pelo número de valores, isto é:

$$x_m = \frac{\sum_{i=1}^n x_i}{n} = \frac{x_1 + x_2 + \dots + x_n}{n}. \quad (2.3)$$

Ou seja, a média aritmética, é a soma das observações dividida pelo número destas.

2.4.2 Média aritmética ponderada

Seja x uma variável quantitativa que assume os valores x_1, x_2, \dots, x_k com frequências absolutas respectivamente iguais a n_1, n_2, \dots, n_k . A média aritmética ponderada de x , indicada por x_p , é definida como a divisão da soma de todos os produtos $x_i \cdot n_i$, com $(i = 1, 2, \dots, k)$ pela soma das frequências, isto é:

$$x_p = \frac{\sum_{i=1}^k x_i \cdot n_i}{\sum_{i=1}^k n_i} = \frac{x_1 n_1 + x_2 n_2 + \dots + x_k n_k}{n_1 + n_2 + \dots + n_k}. \quad (2.4)$$

Lembrando que a frequência relativa (F_i) é definida por $F_i = \frac{n_i}{\sum_{i=1}^k n_i}$, é possível

também expressar a média por:

$$x_p = \sum_{i=1}^k x_i \cdot F_i = x_1 \cdot F_1 + x_2 \cdot F_2 + \cdots + x_k \cdot F_k. \quad (2.5)$$

2.4.3 Mediana

Sejam $x_1 \leq x_2 \leq \cdots \leq x_n$ os n valores ordenados de uma variável x .

A mediana desse conjunto de valores, indicada por Me , é definida por:

$$Me = \begin{cases} x_{\frac{n+1}{2}} & \text{se } n \text{ é ímpar; e} \\ \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} & \text{se } n \text{ é par.} \end{cases} \quad (2.6)$$

Resumidamente, podemos afirmar que, para o cálculo da mediana o conjunto de valores devem estar organizados em ordem crescente ou decrescente. A esta organização denominamos *rol*. E mais, a mediana divide o rol em duas partes com iguais números de valores.

2.4.4 Moda

A moda é definida como a realização mais frequente do conjunto de valores observados.

Seja x uma variável quantitativa que assume os valores x_1, x_2, \dots, x_k , com frequências absolutas iguais a n_1, n_2, \dots, n_k , respectivamente. Se o máximo entre n_1, n_2, \dots, n_k é igual a n_j , $j \in \{1, 2, \dots, k\}$, dizemos que a moda, indicada por Mo , é igual ao valor x_j . Ou seja, a moda de um conjunto de valores corresponde ao valor que ocorre mais vezes.

Exemplo: Retomemos a tabela da Figura 2.1, mais especificamente os dados da coluna, Adj Close, preço do fechamento ajustado do ativo da Magazine Luiza.SA ofertado na bolsa de valores, em reais: 2,58; 2,52; 2,52; 2,35; 2,24. Para calcularmos o preço médio do fechamento ajustado, para esses 5 dias, faríamos:

$$x_m = \frac{2,58 + 2,52 + 2,52 + 2,35 + 2,24}{5} = 2,442.$$

O resultado significa que o preço médio do ativo, após o fechamento ajustado referente a esses 5 dias, foi de aproximadamente R\$ 2,42.

Para calcularmos a mediana, nesse conjunto de dados. Como já estão em ordem decrescente, basta determinar o termo central, visto que possui um número ímpar de elementos. E como neste caso, possui poucos elementos, então é claramente o elemento que ocupa a ter-

ceira posição, ou seja, $Me=2,52$. Porém, com a finalidade de ilustrarmos a maneira que é feito o cálculo para encontrarmos a mediana, num conjunto de dados com o número de elementos ímpar, após escrevê-los em ordem crescente ou decrescente, basta usarmos a expressão:

$$Me = x_{\frac{n+1}{2}} = x_{\frac{5+1}{2}} = x_3 = 2,52,$$

pois 2,52 é o termo que ocupa a terceira posição, após a organização em ordem decrescente, no caso.

Para calcularmos a moda, basta observarmos que a informação mais repetida foi, por coincidência 2,52, com $F_A = 2$.

Imaginemos a seguinte situação, com a finalidade de exemplificarmos o uso da média aritmética ponderada, um investidor compra 10 quantidades desses ativos com o preço de fechamento ajustável: no primeiro, terceiro e quinto dia e repetindo o procedimento para o segundo e quarto dias, mas dessa vez comprando 5 quantidades desse mesmo ativo. Qual seria o preço médio, ao final desses cinco dias, para esse investidor?

$$x_p = \frac{2,58.10 + 2,52.5 + 2,52.10 + 2,35.5 + 2,24.10}{10 + 5 + 10 + 5 + 10} = 2,44375.$$

Ou seja, o preço médio da ação nas condições descritas para o investidor será de aproximadamente R\$ 2,44.

2.5 Medidas de dispersão

Para (DIAS, 2018) as medidas de dispersão expressam um número que representa a variação de um conjunto de dados em relação à média, como é o caso da variância, desvio padrão e coeficiente de variação; ou a variação entre duas variáveis, que mediremos pela covariância e correlação. As definições contidas sobre tais objetos podem ser encontradas em Iezzi, Hazzan & Degenszajn (2013); Bussab & Morettin (2017) e Magalhães & Lima (2023).

2.5.1 Variância

Seja x uma variável quantitativa que assume os valores x_1, x_2, \dots, x_n e x_m a média aritmética correspondente a esses valores. A variância, indicada por σ^2 , é definida por:

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - x_m)^2}{n} = \frac{(x_1 - x_m)^2 + (x_2 - x_m)^2 + \dots + (x_n - x_m)^2}{n} \quad (2.7)$$

Ou seja, a variância é a média aritmética do quadrado dos desvios em torno de sua

média.

2.5.2 Desvio-padrão

Sejam x_1, x_2, \dots, x_n os valores assumidos por uma variável x . Chamamos desvio padrão de x , indicado por σ , a raiz quadrada da variância de x .

$$\sigma = \sqrt{\frac{(x_1 - x_m)^2 + (x_2 - x_m)^2 + \dots + (x_n - x_m)^2}{n}} \quad (2.8)$$

2.5.3 Coeficiente de Variação

Sejam x_1, x_2, \dots, x_n os valores assumidos por uma variável x . O Coeficiente de Variação, que representamos por CV , é a razão entre o desvio padrão e a média do conjunto de dados. Como geralmente é expresso em termos percentuais, fica:

$$CV = \frac{\sigma}{x_m} \times 100\% \quad (2.9)$$

Retomando a Figura 2.1, e sobre a variável preço ajustado após o fechamento, no período supramencionado e levando em consideração que este período se refere a um curtíssimo período, $n = 5$ e com média aritmética já calculada para esse conjunto de dados, $x_m = 2,442$. Vamos calcular a variância o desvio padrão e o coeficiente de variação, obtendo:

- Para variância:

$$\sigma^2 = \frac{(2,58 - 2,442)^2 + (2,52 - 2,442)^2 + (2,52 - 2,442)^2 + (2,35 - 2,442)^2 + (2,24 - 2,442)^2}{5 - 1}$$

$$S^2 = 0,02012.$$

- Para o desvio padrão:

$$\sigma = \sqrt{S^2} = \sqrt{0,02012} \approx 0,1418.$$

- Para o coeficiente de variação, expresso em porcentagem:

$$CV \approx \frac{0,1418}{2,442} \times 100\% \approx 5,81\%.$$

Portanto, para os valores sugeridos, obtivemos a variância igual a 0,02012, o desvio padrão e coeficiente de variação aproximadamente iguais a 0,1418 e 5,81%, respectivamente.

Para medirmos a relação entre duas variáveis iremos usar definições contidas nos trabalhos de Dias (2018), Bussab & Morettin (2017) e Magalhães & Lima (2023) sobre covariância. Para o cálculo desta, as variáveis devem se encontrar na mesma unidade; e o coeficiente de correlação, as variáveis podem não se encontrar na mesma unidade.

2.5.4 Covariância

Sejam x e y duas variáveis quantitativas; x_1, x_2, \dots, x_n e y_1, y_2, \dots, y_n os valores assumidos por x e y , respectivamente. Define-se covariância das amostras, representada por COV_{xy} , da seguinte forma:

$$COV_{x,y} = \frac{\sum_{i=1}^n (x_i - x_m)(y_i - y_m)}{n - 1}. \quad (2.10)$$

Uma outra maneira para expressar a Covariância, entre n-variáveis, é a forma matricial, como segue:

$$\mathbf{Cov}_{x,y} = \begin{bmatrix} var_1 & COV_{12} & \dots & COV_{1n} \\ COV_{21} & var_2 & \dots & COV_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ COV_{n1} & COV_{n2} & \dots & var_n \end{bmatrix}$$

2.5.5 Coeficiente de Correlação

Sejam dois conjuntos de variáveis x_i e y_i , com $i = 1, 2, \dots, n$. O coeficiente de correlação, assim como a covariância, mede o grau de associação linear entre duas ou mais variáveis. Sua simbologia aqui é dada por $corr_{x,y}$, dado por:

$$corr_{x,y} = \frac{COV_{x,y}}{\sigma_x \sigma_y} \quad (2.11)$$

ou na forma matricial:

$$\mathbf{Corr}_{x,y} = \begin{bmatrix} corr_{1,1} & corr_{1,2} & \dots & corr_{1,n} \\ corr_{2,1} & corr_{2,2} & \dots & corr_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ corr_{n,1} & corr_{n,2} & \dots & corr_{n,n} \end{bmatrix}$$

Neste contexto, importante saber que:

- $corr_{x,y} > 0$, significa que as variáveis possuem grau de associação semelhantes;
- $-1 \leq corr_{x,y} \leq 1$, em que, se:
 - $corr_{x,y} = 1$, então as variáveis x e y têm uma correlação linear positiva perfeita;
 - $corr_{x,y} = -1$, então as variáveis x e y têm uma correlação linear negativa perfeita;
 - $corr_{x,y} = 0$, então as variáveis x e y não possuem correlação linear.
- Se, $corr_{x,y}$ resultar em um valor próximo de 1 ou -1 , dizemos que a correlação linear é forte. Caso resulte em um valor próximo de zero, concluímos que a correlação linear é fraca.

Como exemplo, vamos continuar usando as informações da Magazine Luisa.SA, do fechamento ajustado no período de segunda-feira, 18/09/2023 à sexta-feira 22/09/2023 e somando-se a estas as informações da empresa Equatorial Energia.SA, no mesmo período, cujas informações estão na tabela produzida no colab, com dados do site yahoo finance:

Figura 2.6: EQTL3 e MGLU3

[*****100%*****] 2 of 2 completed

	EQTL3.SA	MGLU3.SA
Date		
2023-09-18	33.180000	2.58
2023-09-19	33.279999	2.52
2023-09-20	33.540001	2.52
2023-09-21	33.150002	2.35
2023-09-22	32.880001	2.24

Fonte: do autor

Para calcularmos a covariância entre o preço Adj Close, fechamento ajustado, dos ativos, primeiro, devemos calcular a média aritmética, chamando x de Equatorial Energia.SA e y de Magazine Luiza.SA. Sendo que este último, já foi calculado $y_m = 2,442$. E fazendo

uma média aritmética sobre os valores do fechamento ajustado, obtemos $x_m \approx 33,2060006$. Substituindo os valores encontrados na expressão da covariância, obtemos: $cov_{x,y} \approx 0,02481$.

Com o intuito de usarmos a forma matricial, calculemos a variação entre os ativos, obtendo: $Var_x = 0,05678$ e $Var_y = 0,02012$. E sabendo que, $cov_{xy} = cov_{yx}$, fica:

$$\mathbf{COV}_{x,y} = \begin{bmatrix} Var_x & cov_{xy} \\ cov_{yx} & Var_y \end{bmatrix}$$

$$\mathbf{COV}_{x,y} = \begin{bmatrix} 0,05678 & 0,02481 \\ 0,02481 & 0,02012 \end{bmatrix}$$

Para calcularmos o Coeficiente de correlação entre os ativos, devemos primeiro calcular o desvio padrão de cada um dos ativos, lembrando que para isso, basta encontrar a raiz quadrada da Variância. Portanto, $\sigma_x = 0,23828$ e $\sigma_y = 0,14184$ e o coeficiente de correlação fica:

$$corr_{x,y} = \frac{cov_{x,y}}{\sigma_x \sigma_y} = \frac{0,02841}{0,23828 \cdot 0,14184} \approx 0,73407.$$

Na forma matricial, poderíamos escrever genericamente:

$$\mathbf{CORR}_{x,y} = \begin{bmatrix} corr_{x,x} & corr_{x,y} \\ corr_{y,x} & corr_{y,y} \end{bmatrix}$$

Observe que $corr_{x,x} = corr_{y,y} = 1$, por indicar a correlação entre o mesmo ativo, portanto correlação positiva perfeita. E mais, $corr_{x,y} = corr_{y,x} \approx 0,73407$, conforme já calculado. Substituindo estes valores na matriz, teremos:

$$\mathbf{CORR}_{x,y} = \begin{bmatrix} 1 & 0,73407 \\ 0,73407 & 1 \end{bmatrix}$$

Observe que nas linhas da diagonal principal teremos sempre o valor da correlação igual a +1 pois teremos a correlação entre o mesmo ativo, o que de certo é uma correlação positiva perfeita; o resultado da correlação entre x e y mais próximo de +1 indicaria, um grau de correlação forte. Em verdade, porém, só podemos afirmar que nestes 5 dias, recorte da nossa amostra, os ativos apresentaram comportamento muito próximo, em relação ao

fechamento ajustável.

No próximo capítulo, escreveremos sobre o modelo de Markowitz para a formação de carteiras e este modelo consiste em justificar a opção de investimentos, por intermédio de um modelo matemático, que usa a estatística para isso.

Modelo de formação de carteira

3.1 Conceitos iniciais

Para [Lima \(2016\)](#), o mercado financeiro se caracteriza pela negociação do dinheiro, de modo a intermediar a troca entre empresas que necessitam de capitalização e pessoas dispostas a financiá-las. Segundo [Dias \(2018\)](#), é objetivo da bolsa de valores intermediar tais negociações. No Brasil, a principal bolsa de valores é a Brasil Bolsa Balcão – B3.

Para [Dias \(2018\)](#), os ativos são frações de empresas e são oferecidos nas bolsas de valores e quem compra realiza um investimento que permite ao comprador a participação em ganhos futuros. Segundo [Alves \(2020\)](#), os ativos podem ser classificados como: reais, se estes geram renda líquida para a economia ou financeiros quando alocam os bens para o futuro. Quando os investidores, optam pelo segundo, as empresas podem investir em bens que visam melhorar seu desempenho.

Um investimento, segundo [Dias \(2018\)](#), é uma operação financeira que almeja a obtenção de lucro ou a proteção do capital a fatores como a inflação. Ainda segundo o autor existem dois grupos de investimentos: os de renda fixa, cujas taxas e períodos estão acordadas no momento da aquisição e os de renda variável, onde os retornos não são conhecidos no momento do investimento, podendo inclusive ocorrer perdas.

Para [Iezzi, Hazzan & Degenszajn \(2013\)](#) a inflação é a alta generalizada dos preços de bens e serviços, a consequência é a perda do poder aquisitivo; e a deflação que é a baixa generalizada dos preços de bens e serviços, e como consequência a queda de produção e desemprego.

No Brasil, o principal índice que mede a inflação é o Índice de Preços ao Consumidor Amplo – IPCA, que é calculado comparando os preços dos produtos da cesta básica em relação aos do mês anterior.

Uma outra taxa relevante para o investidor é a do Sistema Especial de Liquidação e Custódia, taxa Selic, que segundo [Dias \(2018\)](#) é obtido pela média ponderada dos juros

praticados pelas instituições financeiras. Esta taxa é utilizada pelo governo como instrumento de controle dos índices inflacionários.

Para [Iezzi, Hazzan & Degenszajn \(2013\)](#), juros é o prêmio cobrado pelo empréstimo de um capital, a uma determinada taxa, por um período, geralmente este último, mensurado em dias, meses, anos etc. Quando um investidor possui mais de um ativo, temos uma carteira de investimentos e que para [Lima \(2016\)](#) existem inúmeras possibilidades para sua formação e em cada uma destas é esperado algum retorno e diferentes níveis de risco.

3.2 Modelo de Markowitz

[Markowitz \(1991\)](#), escreveu o artigo *Portfólio Selection*¹, que propõe as escolhas de ativos baseados em fatores como: o retorno esperado; o risco; e a diversificação dos ativos como medida de redução ao risco. Contrariando os investidores da época, que segundo [Galdino & Rodrigues \(2022\)](#), utilizavam como único critério a escolha de ativos que permitissem o maior retorno, sem considerar o risco. E mais, suas ideias serviram de base para a Moderna Teoria de carteiras.

[Markowitz \(1991\)](#), apoiado em modelos matemáticos, defende os cálculos de retorno; risco; e diversificação, para cada ativo de uma carteira de investimentos, por meio de dados históricos. Com o objetivo de fundamentar a tomada de decisão, segundo as seguintes premissas:

- Os investidores associam risco a variabilidade das taxas de retorno dos ativos em análise. Quanto mais variáveis (voláteis) essas taxas de retorno ao longo do tempo maior o risco de investimento;
- Os investidores avaliam o seu portfólio (carteira) apenas com base no valor esperado e na variância (ou desvio padrão) das taxas de retorno sobre o horizonte de um período;
- Os investidores nunca estão satisfeitos. Quando postos a escolher entre dois portfólios de mesmo risco, sempre escolherão o de maior retorno;
- Os investidores são avessos ao risco. Quando postos a escolher entre dois portfólios de mesmo retorno, sempre escolherão o de menor risco; e
- Os custos de transação e impostos são irrelevantes. ([ALVES, 2020](#), p. 29 - 30)

Para [Lima \(2019\)](#), tais critérios podem ser mensurados por intermédio de conceitos estatísticos, como: média; Variância; desvio-padrão; covariância; e correlação. Tomando em consideração o peso que cada ativo possui na carteira. Para um melhor entendimento, procuramos conceituar e como são medidos os critérios supracitados: Retorno; Risco; e diversificação.

¹Seleção de carteiras, tradução nossa.

3.3 Retorno

Segundo Dias (2018); e Alves (2020), retorno é um valor advindo de uma aplicação financeira, em um intervalo de tempo. Se estes valores são positivos, indicam lucro; e quando negativos, prejuízo. E ainda segundo os autores, Markowitz, classifica o retorno como desejado e o risco como indesejado.

Ross et al. (2022), defendem a porcentagem como a melhor forma de comunicar os retornos, pois desta maneira, seria fácil obtê-los independente do montante aplicado.

Para calcularmos o retorno de um ativo, podemos usar o retorno simples ou o retorno logarítmico.

3.3.1 Retorno Simples

Para calcularmos o retorno, por meio da fórmula do retorno simples - R_t , para um único período, basta fazermos a diferença entre o quociente do preço no período t - P_t e o preço no período anterior a este período - P_{t-1} e 1, como segue

$$R_t = \frac{P_t}{P_{t-1}} - 1. \quad (3.1)$$

3.3.2 Retorno Logarítmico

Para realizarmos o cálculo do retorno, usando a fórmula do retorno logarítmico - r_t , para um único período, basta calcular o logaritmo neperiano de $1 +$ Retorno simples, ou seja,

$$r_t = \ln(1 + R_t) = \ln\left(\frac{P_t}{P_{t-1}}\right). \quad (3.2)$$

3.3.3 Retorno Médio

É a média aritmética dos retornos r_m , nos $t > 0$ períodos. Ou seja

$$r_m = \frac{r_1 + r_2 + \dots + r_t}{t}. \quad (3.3)$$

3.3.4 Retorno Esperado

O retorno esperado de uma carteira - $E(R_c)$, é a combinação ponderada dos ativos que a compõe. Ou seja,

$$E(R_c) = \sum_{i=1}^N R_i \cdot W_i. \quad (3.4)$$

Em que o peso, $0 \leq W_i \leq 1$, representa a proporção investida em cada um dos ativos i ; $W_1 + W_2 + \dots + W_N = 1$, indica que o total de recursos disponíveis está investido; e R_i é o retorno esperado para cada ativo i .

3.3.5 Risco e Diversificação

Para Lima (2016) e Schiroky (2007) o risco é a possibilidade de um determinado investimento não entregar o que se espera devido, por exemplo, a variação do preço do ativo no mercado financeiro. Este movimento também é conhecido como volatilidade. Para medirmos o risco de um investimento utilizamos o cálculo do desvio padrão. O desvio-padrão e o risco são grandezas diretamente proporcionais, ou seja, quanto maior o desvio-padrão, maior o risco.

Ross et al. (2022) classifica os riscos como sistemático e não sistemático:

- [.] Um risco sistemático é aquele que influencia um grande número de ativos, cada um em maior ou menor grau. Como os riscos sistemáticos têm efeitos de forma ampla no mercado, às vezes eles são chamados de riscos de mercado.
- [.] O risco não sistemático afeta um único ativo ou um pequeno grupo de ativos. Como esses riscos são exclusivos de empresas ou ativos individuais, às vezes eles são chamados de riscos únicos ou riscos específicos de um ativo. (ROSS et al., 2022, p. 434)

Um grande exemplo de risco sistemático foi a pandemia do Sars–Covid 19, que além de ceifar a vida de milhares de seres humanos, afetou grande parte dos setores econômicos. Contra esse risco, o sistemático ou de mercado, não há medidas a tomar com o fim de evitá-lo. O risco não sistemático, pelo fato de atingir uma pequena parcela do mercado, para Dias (2018), é passível de ser evitado ou amenizá-lo por meio da diversificação.

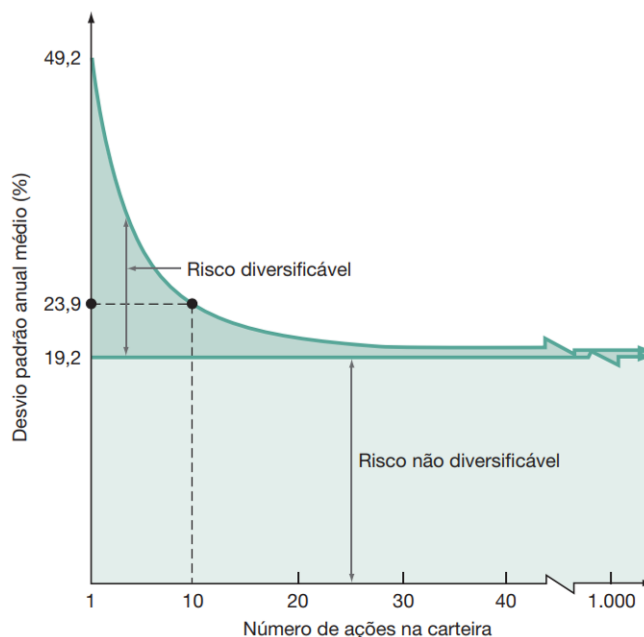
Para Hieda & Oda (1998) a mera distribuição de ativos em uma carteira de investimentos ou mesmo a diversificação ingênua, aquela em que o peso dos ativos é distribuído de maneira igualitária, são medidas que reduzem consideravelmente o risco de um investimento. Sendo que esta última estratégia, é utilizada quando não se tem informações sobre o retorno e risco dos ativos.

Para Markowitz (1991), ao se compor uma carteira de ativos, a busca deve ser a de uma carteira equilibrada em relação ao risco x retorno, de modo que se tenha um máximo retorno para um nível de risco adequado as preferências e objetivos do investidor.

Como vimos, nem todo risco é diversificável, Schiroky (2007), afirma ser impossível,

obter uma carteira de ações livre de risco, ainda que esta seja muito bem diversificada. A Figura 3.1, mostra um exemplo de uma carteira diversificada, por meio do gráfico Desvio padrão x número de ações na carteira.

Figura 3.1: Risco diversificável e não diversificável



Fonte: Ross et al. (2022, p. 437)

No gráfico, podemos notar, que na carteira, existe uma porcentagem que não é passível da diversificação e outra porcentagem, na qual os efeitos da diversificação contribuem para a diminuição de risco. Para medir a diversificação entre dois ou mais ativos, usamos: a covariância e a correlação.

“A Covariância identifica de que forma duas variáveis se relacionam, ou seja, ela avalia como duas variáveis movimentam-se ao mesmo tempo, em relação aos seus valores médios”. (ASSAF NETO, 2008 apud (LIMA, 2016, p. 29)). De acordo com Alves (2020) a covariância mede de maneira eficaz a relação entre dois ativos, por considerar os retornos esperados para os ativos.

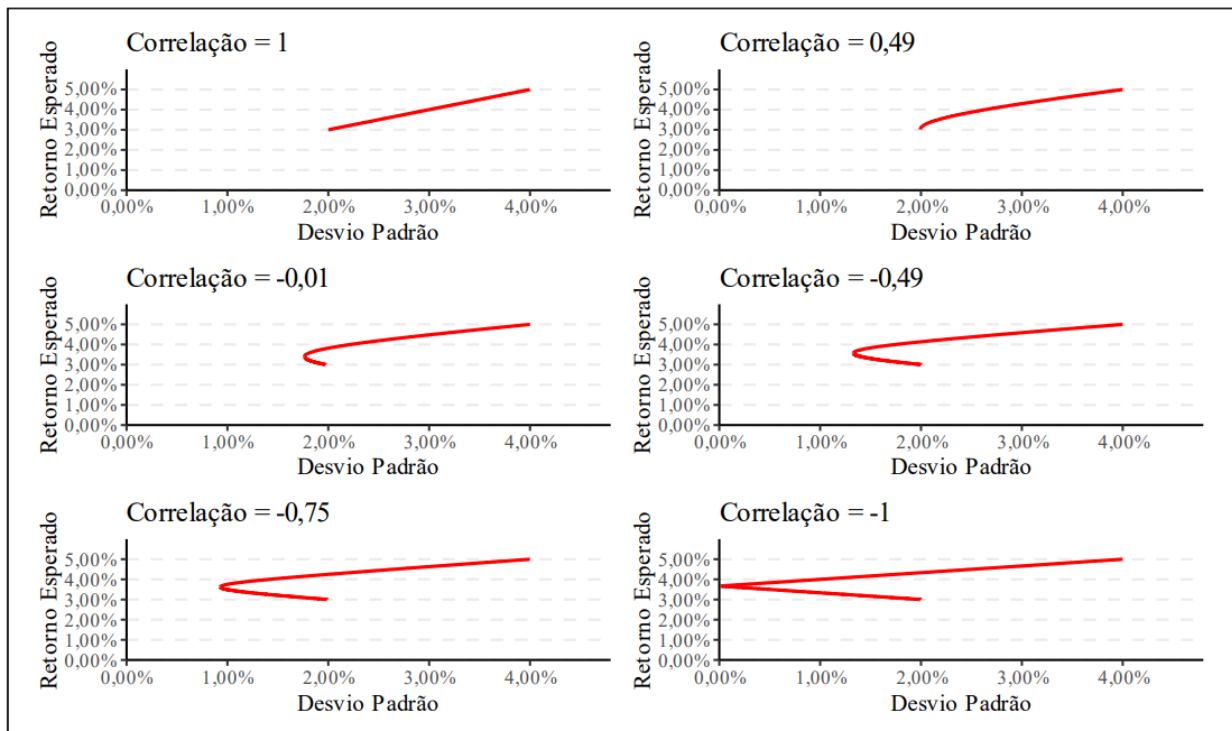
Uma das maiores contribuições de Markowitz, para o campo das finanças foi o de, segundo Galdino & Rodrigues (2022), mostrar que diversificar os ativos não significa possuir muitos ativos na carteira. Mas sim, o de possuir ativos pouco correlatos, pois a variação de preços em um ativo, seria compensada por outro.

Para Dias (2018), Markowitz procurou relacionar ativos com correlações próximos a zero ou negativas com o objetivo de minimizar os riscos em uma carteira de investimentos. E ainda segundo o autor, o comportamento entre os ativos interpretando que o valor da correlação:

- próximo a +1, significa que os ativos possuem os mesmos comportamentos, ou seja, se uma aumenta ou diminui, a outra também o fará;
- próximo a -1, significa que então os ativos possuem relações inversas, ou seja, se uma aumenta a outra diminui, ou vice-versa.
- igual a zero, então os ativos não possuem correlação entre si.

Galdino & Rodrigues (2022), ilustraram o comportamento da correlação simulada entre ativos, por intermédio da Figura 3.2

Figura 3.2: Correlação



Fonte: Galdino & Rodrigues (2022, p. 21)

3.3.6 Risco da carteira

O risco de uma carteira de investimentos é equivalente a calcular a variância desta carteira, que é dada por:

$$\sigma_c = \sum_{i=1}^n \sum_{j=1}^n W_i W_j Cov_{i,j} \quad (3.5)$$

Ou na forma matricial

$$\sigma_c = \sqrt{\begin{bmatrix} w_1 & w_2 & \dots & w_n \end{bmatrix} \cdot \begin{bmatrix} var_1 & cov_{12} & \dots & cov_{1n} \\ cov_{21} & var_2 & \dots & cov_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ cov_{n1} & cov_{n2} & \dots & var_n \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}} \quad (3.6)$$

Onde: σ_c é a variância da carteira; n é o número de ativos; W_i : peso do ativo i na composição da carteira; W_j : peso do ativo j na composição da carteira; e cov_{ij} : covariância entre o par de ativos i e j , se i for diferente de j , e variância se i for igual a j .

3.4 Ativo livre de risco

Encontramos nos textos diferentes versões para qual seria o tipo de investimento, que representaria um ativo livre de risco: para [Hieda & Oda \(1998\)](#) é equivalente à taxa média do Certificado de Depósito Interbancário — CDI; [Schiroky \(2007\)](#), considerou o Título do tesouro direto com vencimento para o dia seguinte; [Galdino & Rodrigues \(2022\)](#), consideraram a poupança ou a taxa do título do Tesouro Direto Pré-fixado. Percebemos, pelos ativos escolhido pelos autores, que são rendimentos de características de investimentos de renda fixa.

Para [Bodie, Kane & Marcus \(2010\)](#), os ativos livre de riscos só podem ser emitidos pelo governo, como é o caso do Tesouro Pré fixado e mesmo este pode haver algum tipo de risco, como por exemplo, no resgate antecipado, onde o preço do título está sujeito a flutuações do mercado.

Segundo [Dias \(2018\)](#) e [Alves \(2020\)](#), o modelo proposto por Markowitz preconizava o conhecimento do risco e retorno para cada ativo da carteira. O objetivo do seu modelo seria o de submeter um conjunto de carteiras a um modelo matemático de modo a encontrar um conjunto de carteiras, as quais para um determinado nível de risco se obtenha um máximo retorno, ou ainda, para um determinado retorno com um risco mínimo. Este conjunto de carteiras é chamada de fronteira eficiente.

3.5 Fronteira Eficiente

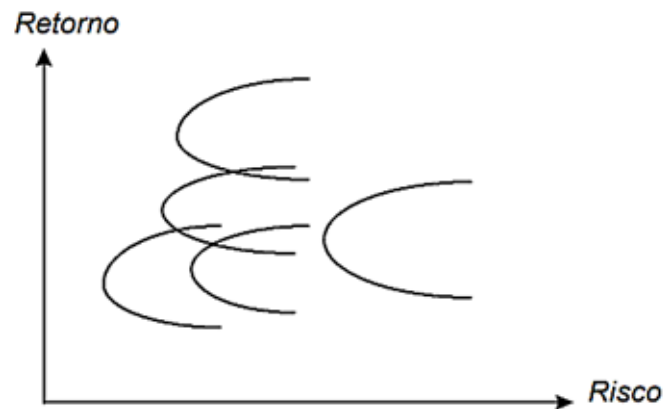
A fronteira eficiente, segundo [Alves \(2020\)](#), é um conjunto de carteiras que sobre a ótica da dicotomia retorno \times risco, visa resolver o seguinte problema:

1. Oferecer o máximo retorno esperado para diferentes níveis de risco; e

2. Oferecer o mínimo risco para diferentes níveis de retorno esperado.” (SHARPE, 1995, p. 194 apud (HIEDA; ODA, 1998, p. 2))

Este problema ficou conhecido como princípio da Dominância. Para Hieda & Oda (1998), Markowitz mostrou que existem infinitas possibilidades de obter os gráficos de eixos Retorno esperado da carteira pelo Risco esperado da carteira. Para dois ativos e variarmos os pesos destes, obteremos um conjunto de hipérbolas. Conforme a Figura 3.3.

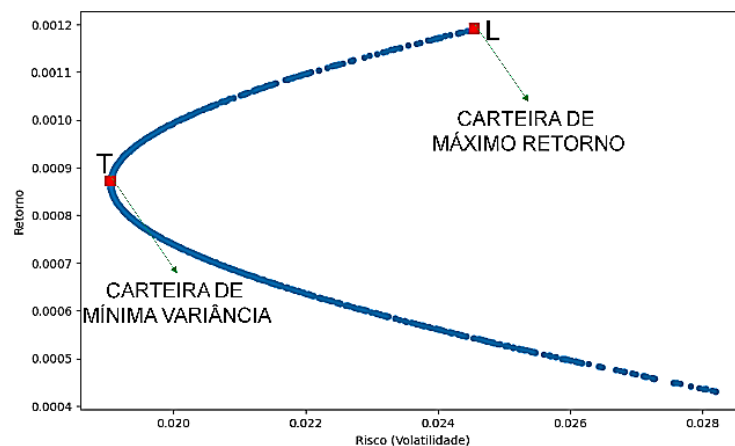
Figura 3.3: Curvas Risco × Retorno



Fonte: Hieda & Oda (1998, p. 3)

Porém, ainda segundo os autores, ao impormos a restrição de cada um dos pesos na carteira pertencerem ao intervalo $[0,1]$ e sua somatória 1, indicaremos que todo recurso disponível está alocado na carteira, e assim o conjunto de carteiras eficientes restringe-se à uma hipérbole. A figura 3.4 é um exemplo do gráfico onde conseguimos identificar a fronteira de eficiência, obtida com os dados de dois ativos, de tickers: TOTS3.SA e VALE3.SA, no período de 01/01/2022 à 01/01/2023.

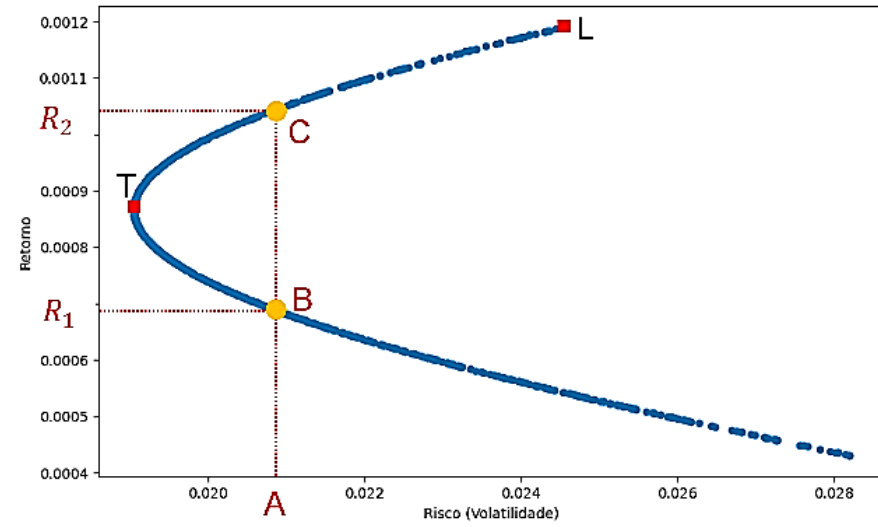
Figura 3.4: Fronteira Eficiente



Fonte: do autor

Notemos que na figura 3.4, a letra T indica a carteira com menor variância ou Carteira de Variância Mínima – CVM; ou seja, a carteira com o menor risco, e L a carteira com maior retorno. Qualquer ponto entre T e L sob a curva é denominado fronteira eficiente. Para uma justificativa, usando o nosso gráfico, imaginemos um investidor que está disposto a correr um risco A conforme a figura 3.5.

Figura 3.5: Princípio da Dominância



Fonte: do autor

Observemos que ao traçarmos uma reta vertical passando pelo ponto A, esta intercepta o gráfico, em dois pontos aos quais denominamos B e C, e destes pontos traçamos duas retas paralelas ao eixo do risco, interceptando o eixo dos retorno em R_1 e R_2 , respectivamente. É fácil ver que, o maior retorno é o R_2 , que parte do ponto C, entre T e L. Portanto, para um risco A, R_2 é o máximo retorno. Repetindo o procedimento, chegaríamos à conclusão de que os maiores retornos, para um determinado risco, estão todos em pontos do gráfico entre T e L, esta é, portanto, a fronteira de eficiência.

Segundo Galdino & Rodrigues (2022), a curva representada pela fronteira de Markowitz, seleciona carteiras com diferentes pesos para os ativos, porém não indica quais as proporções de cada ativo na carteira. Pois, para obter tais proporções deve-se levar em consideração fatores, como: Nível de aversão ao risco do investidor, um ativo livre de risco e a reta que tangência a fronteira eficiente que passe pelo ponto que represente o ativo livre de risco.

3.6 Seleção de Carteiras Ótimas

Para Dias (2018), não existe uma carteira ótima para qualquer tipo de investidor, pois a formação desta carteira depende do perfil de cada investidor, que podem ser classificados, como: Conservador, Moderado e Arrojado.

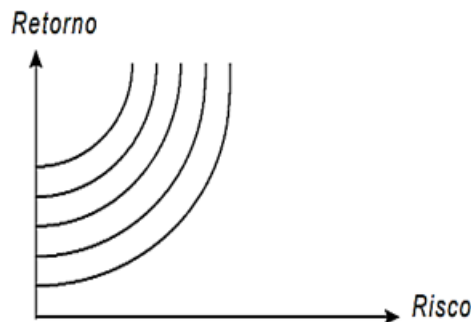
- Conservador - privilegia a segurança e faz todo o possível para diminuir o risco de perdas, para isso aceitando até uma rentabilidade menor.
- Moderado - procura um equilíbrio entre segurança e rentabilidade e está disposto a correr um certo risco para que o seu dinheiro renda um pouco mais do que as aplicações mais seguras.
- Arrojado - privilegia a rentabilidade e é capaz de correr grandes riscos para que seu investimento renda o máximo possível. (ALVES, 2020, p. 27)

Como se vê, o perfil dos investidores exibidos no trabalho de Alves (2020), revela a preferência dos investidores por meio dos critérios encontrados em Markowitz (1952). Para Galdino & Rodrigues (2022), o nível de satisfação com as escolhas do investidor, é uma grandeza individual, que é medido pela função utilidade. Ainda segundo os autores, é definida por Sharpe (1964), como:

$$f(R_c, \sigma)$$

Hieda & Oda (1998), ilustram a função utilidade, conforme as curvas da Figura 3.6, onde curvas mais altas indicam um maior nível de satisfação do investidor.

Figura 3.6: Curvas de Utilidade



Fonte: Hieda & Oda (1998, p. 3)

Ainda segundo os autores, a solução da carteira ótima, ocorre no ponto de tangência entre uma curva de utilidade e a fronteira eficiente. E mais, afirmam que: a formação da fronteira eficiente, não depende do tipo de investidor, e sim do princípio da dominância; enquanto, a solução para o problema de obter uma carteira ótima, depende do tipo de investidor. Ainda segundo a autora, para considerarmos uma carteira ótima, devemos analisar os

critérios adotados. No caso, a maximização da utilidade.

Os investidores maximizadores de utilidade, são aqueles que, “admitindo-se a hipótese de racionalidade dos agentes, os investidores maximizadores de utilidade deveriam procurar conjuntos eficientes de ativos que, maximizem o retorno esperado e, simultaneamente, minimizem o risco corrido” (GALDINO; RODRIGUES, 2022, p. 19).

Nós usaremos, assim como Bodie, Kane & Marcus (2010), a expressão da função utilidade - U , como:

$$U = E(R_c) - 0,5 \cdot A \cdot \sigma^2 \quad (3.7)$$

Matematicamente, a maximização da utilidade, é a expressão obtida, quando fazemos a derivada parcial de U , em relação aos pesos, igual a zero. Obtendo:

$$w = \frac{E(R_C) - R_F}{A\sigma^2} \quad (3.8)$$

Em que: U é a função utilidade; R_e é o retorno esperado da carteira; A é o nível de aversão ao risco; e σ^2 é a variância.

Geralmente, os bancos usam questionários que identificam o perfil dos investidores, em que se estima o nível de aversão ao risco, para propor aos clientes produtos condizentes com o seu perfil. Como o questionário do Banco Paulista, que consta no anexo deste trabalho.

3.6.1 Índice de Sharpe

Influenciado pelas ideias de Markowitz, segundo Galdino & Rodrigues (2022), Sharpe em 1964 também conclui, a respeito da avaliação dos ativos de uma carteira de ativos, os mesmos critérios, e chegou em uma expressão que relacionou o retorno de um ativo livre de risco, um ativo de risco e o desvio padrão. Tal expressão ficou conhecida por índice de Sharpe. Segundo Schiroky (2007), quanto maior o índice de Sharpe maior será o retorno e menor será o risco.

$$IS_P = \frac{R_P - R_F}{\sigma_p} \quad (3.9)$$

Em que, IS_P é o Índice de Sharpe; R_P é o Risco do portfólio no ponto P; R_F é o retorno do ativo livre de risco; $R_P - R_F$ é a rentabilidade da carteira ou prêmio pelo risco e σ_p é o risco ou desvio padrão do prêmio.

De acordo com Lima (2019), o índice de Sharpe oferece um critério de escolha por parte do investidor, por representar a maior unidade de retorno por risco global. Ou seja, se o investidor tiver opções, deverá fazer a opção pelo de maior Índice de Sharpe. E mais, o “Índice de Sharpe se baseia na Linha de Mercado de Capitais (Capital Market Line – CML), sendo válido na avaliação de carteiras pouco diversificadas, em que quanto maior o seu valor, melhor é o desempenho da mesma” (LIMA, 2019, p. 28).

3.6.2 Capital Market Line – CML

Para Schiroky (2007) a CML é obtida, quando tomamos uma carteira de ativos, com dois tipos de ativos: um ativo livre de Risco – R_F e ativo(s) com risco – R_p , de modo que os pesos desses ativos - w_i seja igual a 1. O retorno esperado por essa combinação será representado pela expressão $E(R_C)$:

$$E(R_C) = w \cdot R_P + (1 - w) \cdot R_F \quad (3.10)$$

Já para calcularmos, o desvio padrão, o faremos pela expressão:

$$\sigma_C = \sqrt{(1 - w)^2 \sigma_F^2 + w^2 \sigma_P^2 + 2w(1 - w) \sigma_P \sigma_F \text{Cov}_{FP}} \quad (3.11)$$

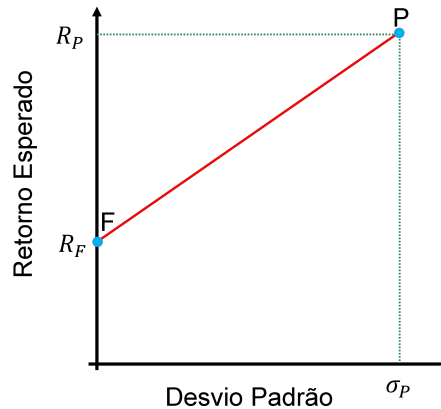
Como o ativo livre de risco, não sofre os efeitos da volatilidade, $\sigma_F = 0$, resultando em:

$$\sigma_C = w \sigma_P \implies w = \frac{\sigma_C}{\sigma_P} \quad (3.12)$$

Substituindo 3.10 em 3.12, obtemos:

$$E(R_C) = R_F + (R_P - R_F) \cdot \frac{\sigma_C}{\sigma_P} \quad (3.13)$$

Segundo Galdino & Rodrigues (2022), quando fazemos as combinações possíveis para os pesos da carteira de um ativo livre de risco e uma carteira de ativos, obtemos a CML, que é a reta composta por um conjunto de carteiras eficientes de ativos com risco e sem risco, conforme podemos visualizar na Figura 3.7.

Figura 3.7: Capital Market Line

Fonte: do autor

Baseado em [Schiroky \(2007\)](#), se as proporções escolhidas pelo investidor estão localizadas no ponto F, então todo o capital está alocado no ativo sem risco; pontos alocados próximos a F, configuram um maior peso na carteira para ativos livre de risco; o ponto médio do segmento FP, mostra que a carteira possui pesos iguais para os ativos livre de risco e com risco; pontos próximos de P, indicam que os pesos da carteira em ativos de risco é maior do que o ativo sem risco; e se os pesos dos ativos com risco e sem risco, estão indicados pelo ponto P, então os recursos disponíveis estão investidos na carteira de ativos de risco, esta carteira também é conhecida como carteira de mercado.

[Galdino & Rodrigues \(2022\)](#) afirmam que a inclinação da reta é dada pelo Índice de Sharpe. Esta afirmação é de fácil verificação do ponto de vista matemático, basta traçarmos uma linha paralela ao eixo do desvio padrão, interceptando o eixo do retorno esperado em R_F e a reta $P\sigma_P$, em S, tal que o triângulo FSP é retângulo em S, e $PFS = \theta$, conforme ilustramos na Figura 3.8.

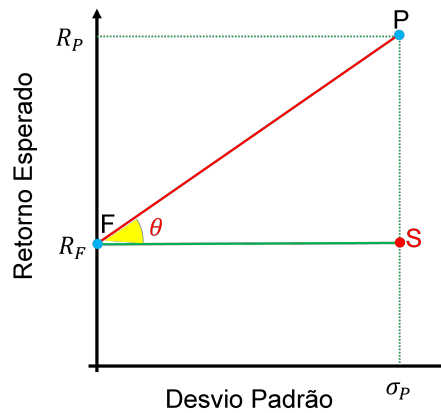
E mais, tomando $F = (0, R_F)$; $P(\sigma_P, R_P)$; $m = tg\theta$; $SP = R_P - R_F$; e $SF = \sigma_P$. Pelas relações métricas no triângulo retângulo, temos:

$$tg\theta = \frac{R_F - R_P}{\sigma_P} \implies m = \frac{R_F - R_P}{\sigma_P} \implies m = IS_P.$$

[Hieda & Oda \(1998\)](#); [Galdino & Rodrigues \(2022\)](#), afirmam que o ponto P, é tal que a reta FP, tangencia a fronteira eficiente em P, em que implicaria que P, é o máximo índice de Sharpe, na fronteira. Conforme ilustrado na Figura 3.9.

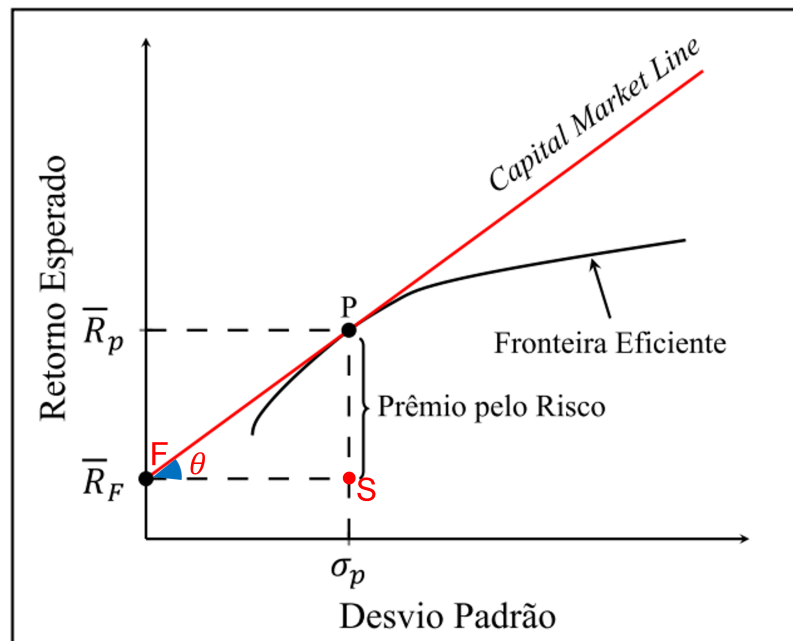
A solução do problema de formar carteiras ótimas, estão todas sobre a CML, sendo que a composição dos pesos de ativos com risco e sem risco, dependem das preferências do investidor. Ou seja, a linha da CML, é também a linha que deve ser proposta aos investidores maximizadores de utilidade. E ainda, quando 100% do capital disponível está sobre a carteira,

Figura 3.8: Equação da reta



Fonte: do autor

Figura 3.9: Carteira Ótima



Fonte: do autor, baseado em Galdino & Rodrigues (2022).

as carteiras ótimas estão entre os pontos F e P, incluindo estes. A relação entre os pesos dos ativos da carteira que ficam neste intervalo, já foi descrita neste texto. Restando apenas a relação dos pontos a direita de P, segundo [Galdino & Rodrigues \(2022\)](#), ocorrem quando se investe uma quantidade maior do que os recursos disponíveis, onde o investidor toma empréstimos com taxas iguais às do ativo livre de risco, com o objetivo de investir na carteira, este processo é conhecido como alavancagem.

4.1 Noções iniciais

Ao perguntarmos a um padeiro, como fazer um pão, ou a um mecânico como se troca o óleo; esperamos obter uma sequência de instruções, que ao seguirmos, nos torna capaz de realizar estas tarefas, desde que, tenhamos as ferramentas necessárias e alguma habilidade para sua execução. Um programador, deve ter a habilidade de se comunicar com a máquina, por meio de um conjunto de instruções que seja inteligível a esta.

Menezes (2019) enumera diversos motivos para programar, fazendo analogia com atividades que fazem parte do cotidiano das pessoas como: cozinhar; aprender a usar mapas; acertar o despertador, entre outras. Essas ações pressupõe conhecimento necessário para seguir um conjunto de regras ou comunicar a máquina que os siga, e a este conjunto denominamos programa.

Uma Linguagem de Programação – LP, para Leite (2020), deve transmitir um conjunto de instruções ao computador, para resolver um determinado problema. Podemos classificar as LP's, como: de baixo nível, quando as instruções dadas são inteligíveis para o processador; enquanto as de alto nível são de fácil compreensão ao programador.

Um código fonte, segundo Silva (2021), é um conjunto de instruções para serem executadas pela máquina por intermédio de um tradutor ou compilador, que possui a funcionalidade de traduzir a linguagem do programador para a linguagem da máquina.

4.2 Noções de *Python*

A linguagem *Python*, conforme afirma Wazlawick (2018), foi concebida por Guido Van Rossum em 1990, matemático e programador holandês; muito embora seja erroneamente associado a cobra pítão, o nome foi inspirado no programa humorístico inglês *Mont Python*. Tal erro acabou influenciando na construção da logomarca da linguagem de programação.

Segundo Menezes (2019), o Python é um programa que possui uma linguagem simples, clara e concisa; de *software* livre, gratuito; possui versão para a maioria dos computadores; e de código aberto, ou seja, permite a colaboração de uma gama considerável de pessoas, isso tem auferido ao Python um número considerável de bibliotecas que permitem solucionar problemas em diversas áreas do conhecimento, como por exemplo, da matemática. Atualmente, encontra-se no topo das linguagens de programação mais usadas no mundo, segundo o Índice Tiobe (ver figura 4.1 abaixo):

Figura 4.1: *Ranking* Tiobe

Janeiro 2023	Janeiro 2022	Linguagem de programação	
1	1		Python
2	2		C
3	4		C++
4	3		Java
5	5		C #

Fonte: <https://www.tiobe.com/tiobe-index/> em 15/02/2023

Dentre algumas opções de *Integrated Development Environment* – IDE¹, escolhemos o *Google Colaboratory* (*Colab*), por ser uma interface de fácil uso, e acessada a qualquer pessoa que possua uma conta *gmail*, em geral não institucional. Além da facilidade de acesso, o *Colab* nos dá a possibilidade de acessar diversas bibliotecas, como por exemplo a *yfinance*, a qual nos permite obter dados da bolsa de valores, com os quais fizemos a aplicação do modelo de Markowitz para formação de uma carteira.

4.2.1 Notebook *Colab*

O Python possui diversas versões *online* e *offline*, no entanto julgamos mais adequada ao nosso trabalho a IDE *notebook colab*. Ele nos permite também deixar o *link* de códigos para que outras pessoas possam lê-los, copiá-los, ou mesmo editá-los mediante compartilhamento. De posse do *gmail* pessoal, para utilizar basta seguir os seguintes passos:

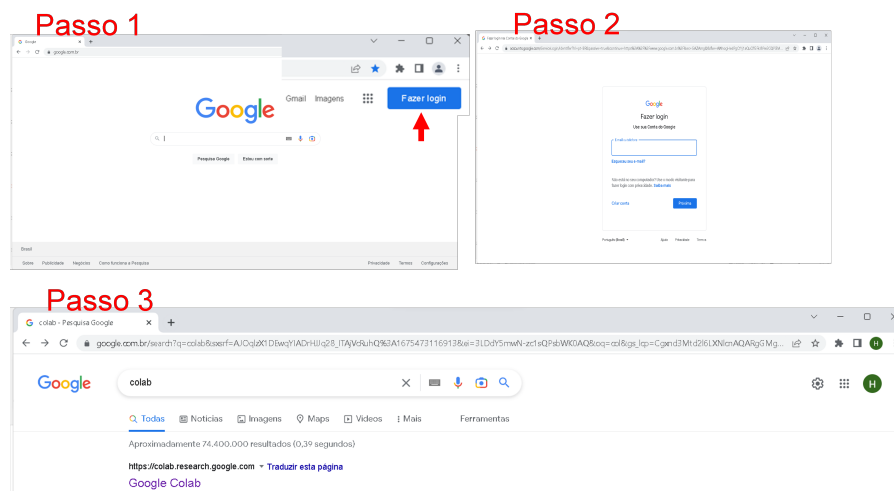
- Passo 1: Selecionar a opção fazer login na plataforma *google*;
- Passo 2: entrar com as credenciais da conta *google*; e

¹*Integrated Development Environments* (IDEs) são pacotes de software que integram várias ferramentas de desenvolvimento em um ambiente consistente com o objetivo de aumentar a produtividade do desenvolvedor. Geralmente as IDEs incluem recursos como *syntax highlight* (código fonte colorizado conforme a sintaxe da linguagem) e acesso fácil a sintaxes de comandos, botões de compilação e depuração (BORGES, 2014, p. 18).

- Passo 3: buscar o *notebook Colab*.

Conforme descrito nas imagens da Figura 4.2:

Figura 4.2: Ilustração do Tutorial de acesso ao *Colab*



Fonte: do autor

Opcionalmente, o acesso ao *Colab* a partir da conta do *gmail* é feita na área do *google drive*. Ali, caso não se tenha ainda construído qualquer código do *Colab*, procedemos com os seguintes passos: [1]: + Novo -> [2]: Mais -> [3]: + Conectar mais apps -> [4]: digitar *Colaboratory* no campo de busca² -> Instalar *Colaboratory*. Isso fará com que da próxima vez o *Google Colaboratory* já apareça em seguida o passo [2].

Ao optarmos pelo acesso por intermédio do https://colab.research.google.com/?utm_source=scs-inde iremos nos deparar com uma tela em que há a opção de criarmos um novo *notebook*, ou caso já tenha algum programa produzido, os arquivos irão ser disponibilizados nesta tela, como podemos perceber na Figura 4.3.

Pela figura 4.3 também é possível observar que os programas confeccionados no *Colab* ficam armazenados em nuvem (no *google drive*), em formato *.ipynb*³. Isto na prática nos permite o acesso, a qualquer programa confeccionado no *Colab*, pelo *google drive* a partir de qualquer primeiro programa criado.

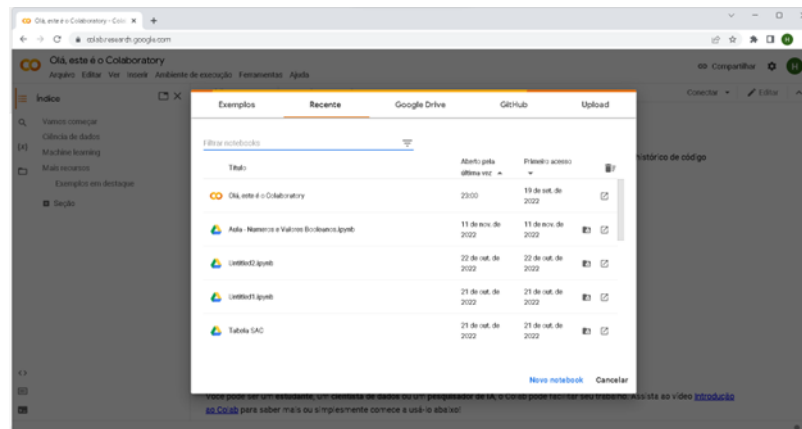
Ao escolhermos a opção **Novo Notebook**, teremos a interface do *notebook Colab*, conforme a Figura 4.4:

Sendo o ambiente de execução bem intuitivo, não há necessidade de descrevermos muitas funcionalidades, porém ressaltamos algumas, de modo a facilitar ações que julgamos importantes.

²Caso não apareça o *Colab* aqui, esse *gmail* não terá acesso a ele.

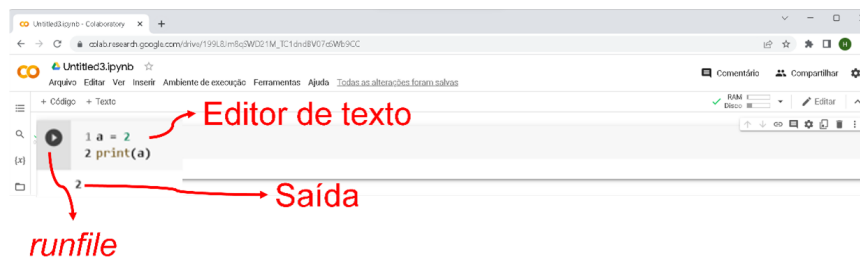
³Extensão do *Python* no *Colab*. Em outras plataformas somente *.py* é a extensão de programas *Python*.

Figura 4.3: Ilustração da página de Documentos da *Google Drive*



Fonte: do autor

Figura 4.4: Ilustração do Ambiente *Colab*

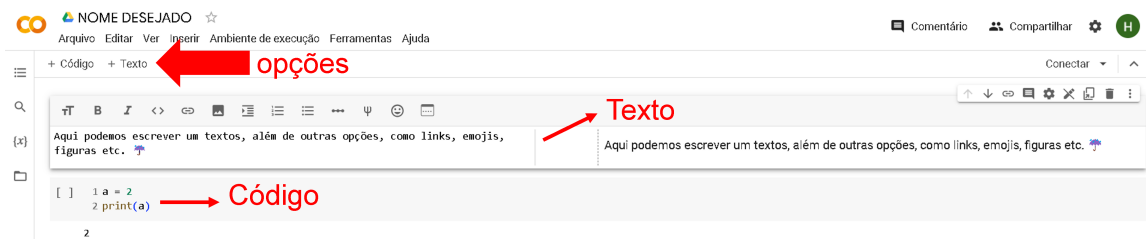


Fonte: do autor

Confeccionando o documento

Para além dos programas, um documento confeccionado no *Colab*, pode conter textos, em diferentes formatos, entre eles inclusive o $\text{L}^{\text{A}}\text{T}^{\text{E}}\text{X}^4$, conforme ilustra a Figura 4.5:

Figura 4.5: Ilustração de inserção de texto e código do *Colab*



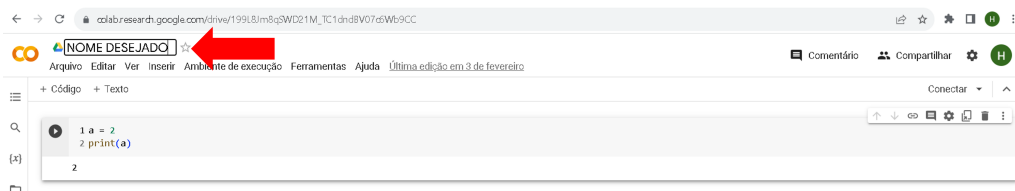
Fonte: do autor

⁴Clicando em ψ , ou escrevendo entre um par de cifrões!

Renomeando um notebook

Os documentos digitados no *notebook*, são salvos no *google drive* automaticamente, isto é, sem que haja necessidade de um comando específico. O nome do arquivo vem no padrão: *Untitled.ipynb*, que para renomeá-lo basta clicar sobre a caixa que o possui (como indica a Figura 4.6), mas não se deve alterar a extensão *.ipynb*.

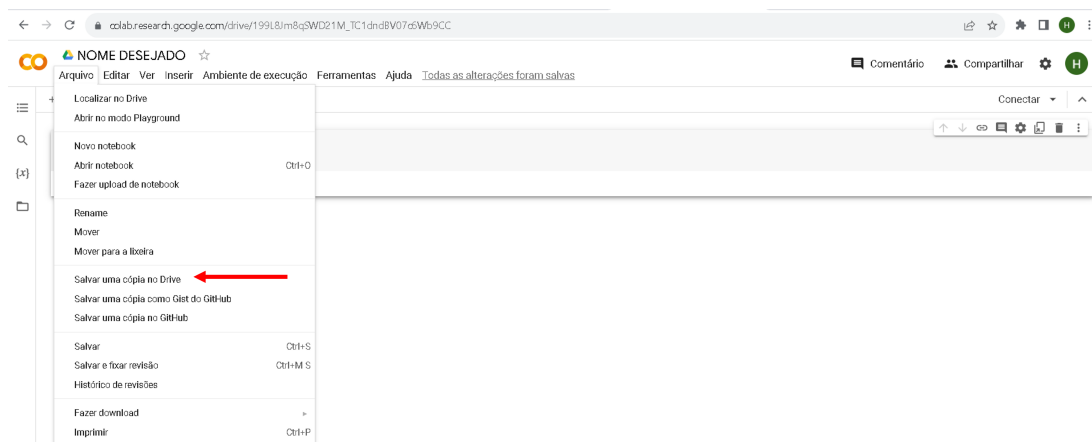
Figura 4.6: Ilustração de como renomear título de *Notebook* do *Colab*



Fonte: do autor

Para realizarmos a cópia de um programa no *Colab*, basta acessar o menu arquivo na barra de ferramentas, logo abaixo do nome do arquivo. Ao seguir este protocolo observe que há muitas outras opções disponíveis neste ícone. Conforme percebemos na Figura 4.7.

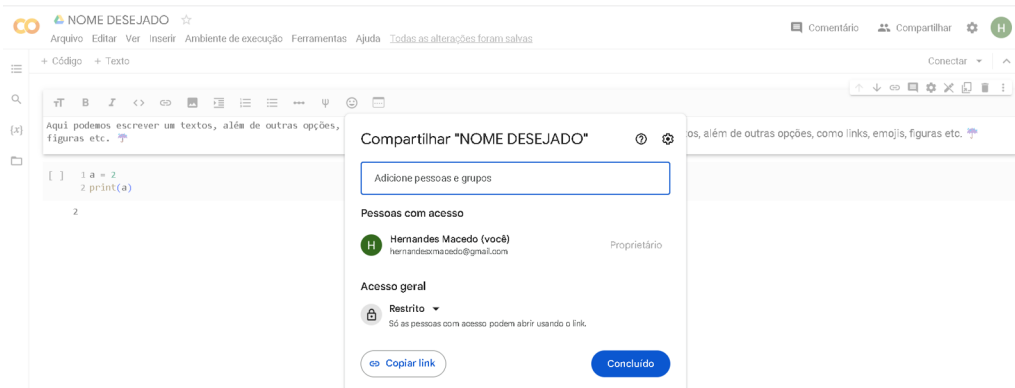
Figura 4.7: Ilustração de como copiar *Notebook* do *Colab*



Fonte: do autor

Compartilhando um programa

Uma outra ação bastante útil no *Colab*, é o de compartilhamento do programa, por meio do *e-mail* ou via *link*, pois além de socializar um programa, permite a edição deste, o que confere uma possibilidade de interação. Isso está ilustrado na Figura 4.8

Figura 4.8: Ilustração de como compartilhar *Notebook* do Colab

Fonte: do autor

4.3 Primeiros passos

4.3.1 print e comentário (#)

Podemos dizer que o comando `print`, palavra que podemos traduzir como imprimir, pede ao programa o que deve ser expresso na saída. Por exemplo, nos trabalhos de [Borges \(2014\)](#); e [Menezes \(2019\)](#) encontramos a saudação olá mundo como exemplo de primeiro programa. Para que esta frase surja na saída, ela deve estar escrita entre aspas, ou entre apóstrofes. Já o comando `#` é usado para escrever algum comentário e estes serão ignorados pelo programa no instante da execução, conforme podemos perceber no Código 4.1:

Código 4.1: Exemplo de uso do `print` e `#`

```

1 #esse comentario não será lido
2 #Vamos usar o comando print
3 print('Olá mundo')
4 print("Olá mundo")

```

O comando `print`, também pode ser substituído pelo comando `display` no *Colab*, e será usado ao longo do texto. Uma outra informação relevante no uso do *Colab*, é que ao guardamos um valor para uma variável, se quisermos visualizá-la basta escrever o nome atribuído a variável, sem usar `print` ou `display`, como exemplificado no Código 4.2:

Código 4.2: Mostrando resultados sem preciso escrever `print`

```

1 #não é preciso escrever print
2 a = 2
3 a

```

O comando `print` também aceita outras sintaxes para inserção de números ou mesmo variáveis dentro de um texto, como será visto mais adiante.

4.3.2 Variáveis

Menezes (2019) faz a diferença de variável em matemática, que é o valor que torna a sentença verdadeira; enquanto que na programação, ao atribuímos um nome a uma variável a máquina entende que um espaço deve ser reservado na memória para o armazenamento de dados. Os principais tipos de variáveis são:

Numéricas:

int variáveis cujos valores pertencem ao conjunto dos números inteiros;

float variáveis que admitem valores representando um número decimal, o qual mesmo se o número for irracional, será representado finito por aritmética de ponto-flutuante; e

complex números complexos com partes real e imaginária sendo pontos-flutuantes. No *Python*, se $z = a + bi$, com $i = \sqrt{-1}$, e $a, b \in \mathbb{R}$ variáveis, ele será armazenado na sintaxe `z=complex(a,b)` e visto na forma `a+bj`. Mas `z=2+3j` é um exemplo de sintaxe de variável complexa quando diretamente queremos escrever os valores das partes real e imaginária.

Uma observação importante é que variáveis do tipo *float*, devem ser representadas por pontos e não vírgulas, que é o nosso padrão.

Para estes tipos de variáveis, segundo Borges (2014), podemos usar operações aritméticas básicas cujos operadores estão descritos na Tabela 4.1:

Tabela 4.1: Operações aritméticas básicas

Operador	Operação
$a + b$	Adição
$a - b$	Subtração
$a * b$	Multiplicação
a / b	Divisão
$a // b$	quociente da divisão inteira de a por b
$a \% b$	resto da divisão inteira de a por b
$a * * b$	Exponenciação ou Potenciação

Fonte: do autor.

Strings e input associando-se a variáveis

Uma *string* armazena caracteres e é atribuída simplesmente colocando entre um par de apóstrofos ou de aspas. Segundo Leite (2022), geralmente é precedida pelo comando `input`, que solicita ao usuário inserir o ‘valor’ da variável. O comando `input` naturalmente recebe *strings*, e sendo necessário usar externamente a ele, `int`, `float`, `complex` ou `bool` para interpretar a entrada como sendo do tipo de variável adequada. Uma coisa interessante é que, diferente da maioria das linguagens de programação, no *Python* se aceita, apesar de não recomendar, o uso de acentos nas declarações de variável.

Podemos tomar como exemplo de adequação do `input` a um tipo de variável específica, o programa do código 4.3, que busca calcular o montante a juros compostos de um investimento realizado a uma única aplicação.

Código 4.3: Exemplificação de usos de `inputs` com `float` e do `print`

```

1 # Programa para calcular o montante de uma unica aplicacao de juros compostos,
   onde:
2 # M: montante;
3 # C: capital
4 # i: taxa
5 # t: numero de períodos
6 C = float(input('Entre com o capital a ser investido: '))
7 i = float(input('Entre com a taxa em porcentagem: '))
8 t = int(input('Entre com o numero de periodos: '))
9 M = C * (1 + i/100)**t
10 print('O montante proveniente de um investimento de R${0},00 a taxa mensal de
      {1}% durante {2} meses é R${3},00'.format(C, i, t, round(M,2)))

```

Note que, ao pedir que o usuário insira as informações do problema, o comando `input` acaba por permitir uma maior interação do usuário com a máquina.

Observe ainda que usamos juntamente com o `print`, o comando `.format` sobre um quarteto ordenado, em que C ocupou a posição 0, i a posição 1, t a 2 e a posição 3 ocupada pela variável M , que foi arredondada para duas casas decimais com o comando `round`.

Booleanas

Variáveis do tipo booleanas são simplesmente valores lógicos `True` (verdade) ou `False` (falso).

Neste caso, baseados em Borges (2014), são comumente resultados nas operações de comparação, cujos operadores básicos são descritos na Tabela 4.2, ou mesmo de atribuição de uma variável dada pelo programador no objetivo de controle de fluxo de tarefas dentro dos programas.

Tabela 4.2: Tabela de operações lógicas

Operador	Operação
<	Menor
>	Maior
<=	Menor do que ou igual
>=	Maior do que ou igual
=	Igual
!=	Diferente

Fonte: do autor.

4.3.3 Estruturas com sentenças condicionais: `if-elif-...-else`

As variáveis booleanas aparecem naturalmente na verificação de sentenças lógicas e a partir delas podemos realizar decisões para tomarmos uma ou outra ação. E quando certas tarefas devem ser executadas somente quando um conjunto de condicionais seja satisfeito, usamos comando de seleção condicional, cuja estrutura geral é:

```

if condicional_1:
    grupo_comandos_1
elif condicional_2:
    grupo_comandos_2
    :
elif condicional_n_1:
    grupo_comandos_n_1
else:
    grupo_comandos_n

```

A estrutura desse controle de fluxo é acompanhada obrigatoriamente de uma sentença lógica a ser verificada em cada `if` e `elif` utilizado. `if` diz para o programa verificar **Se** a sentença `condicional_1` for verdadeira, **então** o `grupo_comandos_1` deve ser executado. Já o `elif` significa **senão se**. Isto é, se a condicional anterior não foi verdadeira, mas se `condicional_?`, que está ao lado do `elif` for, **então** o `grupo_comandos_?` associado deverá ser executado. Ou seja, essa estrutura computacional representa várias implicações lógicas condicionais, que os matemáticos/lógicos usam na argumentação de proposições. O **Se ... então** no *Python* fica representado pela sequência `if` ou `elif` sobre `condicional_?`, com o **então** simplesmente representado pelo símbolo de `:`.

A linha de comando `else:` não é obrigatória (nem mesmo qualquer `elif` é!), e compete o programador decidir se seu uso é ou não necessário. No entanto, usá-lo significa **Senão** para todas as condicionais utilizadas anteriormente. Ou seja, **se** nenhuma das condicionais

verificadas antes forem verdadeiras, **então** execute `grupo_comandos_n`. Novamente, o **então** aqui é indicado pelo símbolo `:`.

Obrigatoriamente deve haver indentação (obtido por um TAB ou quatro espaços simples) na(s) linha(s) que representa(m) o(s) `grupo_comandos_?`. Caso isso não seja feito, a execução não é realizada! Outra obrigatoriedade que se deve ter em mente é que, já que um grupo de comandos é executado quando estão com indentação após a verificação lógica da estrutura, então qualquer comando que estiver fora de indentação não mais pertencerá à estrutura condicional. Ou seja, não mais estará sob avaliação do `if-elif-...-else`. Isso deve ser lembrado até mesmo com essa estrutura sendo encadeada por outra do mesmo tipo, ou com de *loop*, que veremos a seguir!

Já que a estrutura `if-elif-...-else` representa implicações lógicas, podemos sobre as condicionais utilizar de operadores da lógica, como a conjunção ($e = \wedge$), a disjunção (ou $= \vee$) e a negação (\sim). Nesses casos, temos respectivamente, os conectivos `and`, `or` e `not`. Para seus usos é importante lembrar que:

- `p or q`: assume valor lógico verdadeiro quando ao menos uma das sentenças (`p` ou `q`) for verdadeira;
- `p and q`: assume valor lógico verdadeiro somente quando ambas sentenças são verdadeiras;
- `not p`: inverte o valor lógico da sentença `p`.

Como simples exemplo da estrutura com sentença condicional construímos o programa 4.4 baseado no cálculo da média anual dos alunos do ensino regular da rede pública do estado do Pará, que é realizado segundo a média ponderada de pesos 2, 3, 2, 3, referente às 1^a, 2^a, 3^a e 4^a avaliações, respectivamente, e a média a ser obtida para aprovação é 5.0.

Código 4.4: Exemplo de uso `if` e `else`

```
1 #não esqueça de ao entrar com a nota, escrever ponto no lugar da vírgula
2 nota1=float(input('entre com a nota do primeiro bimestre: '))
3 nota2=float(input('entre com a nota do segundo bimestre: '))
4 nota3=float(input('entre com a nota do terceiro bimestre: '))
5 nota4=float(input('entre com a nota do quarto bimestre: '))
6 média = (nota1*2+nota2*3+nota3*2+nota4*3)/10
7 print(média)
8 if média >= 5.0: #não esquecer de finalizar com :
9     print('Aprovado') #não esquecer dque essa linha deve ser indentada
10 else:
11     print('Reprovado')
```

4.3.4 Estruturas de repetição: `while` e `for`

Das leituras de Paiva et al. (2021); Borges (2014); e Menezes (2019), tanto o `while`, quanto o `for` são estruturas usadas quando se quer realizar operações repetitivas, também conhecidas como *loop*. As operações repetitivas ficam indentadas, e a “quebra” da indentação significa o fim das repetições.

Quando certas tarefas não possuem previamente um quantidade definida de vezes que deve ser realizada, e sim que seja executada até que uma certa condição seja satisfeita; teremos necessidade de usar um *loop* condicional. No *Python* a estrutura para isso é o `while`, cuja sintaxe é:

```
while condicional:  
    grupo_comandos
```

Ele significa que **enquanto** a *condicional* for verdadeira, execute `grupo_comandos`. Assim, essa estrutura não possui declaradamente um *contador* do número de repetições (iterações) que foram processadas na execução. Um *contador* serve para fazer o que seu nome diz, e que em geral possui uma necessidade posterior em se conhecer. Um pequeno exemplo de uso do `while` está no código 4.5 abaixo, o qual simplesmente mostra todos os números inteiros de 0 a 10.

Código 4.5: Exemplo de uso do `while`

```
1 #Programa que escreve os inteiros no intervalo [0, 10]  
2 x = 0  
3 while x <= 10:  
4     print(x)  
5     x = x + 1
```

Um comando que é executado enquanto $x \leq 10$ é o de “printar” o valor de x . Mas, como inicialmente $x = 0$, sua apresentação seria infinita caso não existisse uma modificação de seu valor. Daí, com o propósito de mostrar os números de 0 a 10 com o `while`, houve a necessidade de escrever x como sendo de um “acumulador”, que, neste caso, acrescenta ao valor anterior uma unidade. Ou seja, o valor de x em cada iteração é modificado para ser o valor anterior mais 1. É importante salientar aos iniciantes que $x = x + 1$ ⁵ não é uma equação (nem poderia ser!), e tal escrita é a forma computacional de se atualizar uma variável que depende da quantidade de vezes que acontece repetição.

⁵Uma forma equivalente de $x=x+1$ no *Python* é $x += 1$. O mesmo acontece para qualquer outra operação de acumulação, com $*$, $/$, $-$ ou $**$, por exemplo!

No exemplo acima `x` também serviria de contador, pois quando `x` for 11, ele não mais seria mostrado, mas representaria o número de vezes (e de algarismos) que o *loop* foi executado!

Obrigatoriamente devemos usar a indentação após a linha da condicional do *loop while*, para se configurar como o conjunto de comandos a serem executados sob a condicional ser verdadeira. E lembrar que, retirando-se tal indentação associada o *loop*, não mais teremos comandos de repetição desse *while*!

Quando desejamos que uma certa tarefa seja realizada por um número n finito de vezes usamos a estrutura de repetição *for*, cuja forma configura-se como:

```
for variavel_contador in dominio_iteracao:
    comando_iteracao
```

Sob essa estrutura, o que devemos interpretar e lembrar é que:

- A `variavel_contador` serve para identificação do elemento ocorrido em cada iteração, a qual pertencerá (`in`) ao `dominio_iteracao` (às vezes não todos elementos);
- Não esquecer dos dois-pontos no final da primeira linha;
- Deve ocorrer a indentação para cada `comando_iteracao`;
- Qualquer comando sem indentação não mais pertencerá ao *loop for*.

O elemento `dominio_iteracao` pode ser um lista (elementos dispostos entre colchetes e separados por vírgulas) numérica ou de *strings*, uma variável *string*, ou linhas e colunas de uma tabela (como veremos no capítulo 5). No entanto, para exemplificação aqui, lista numérica é que será usada; e nesse caso, para uma sequência numérica crescente, o comando `range(ini,fin,inc)` faz com que `variavel_contador` percorra uma lista cujo primeiro elemento é `ini`, vai até `fin-1`, com incremento entre os elementos sendo `inc`. Já para um sequência decrescente, seria necessário `ini` ser maior que `fin` e `inc` ser negativo. Além disso, o último elemento terá relação com o número `fin+1`. Também é importante pensar que, dependendo de `inc`, pode ser que `fin-1`, ou `fin+1` não seja alcançado, já que os elementos serão dispostos a partir de `ini` de `inc` em `inc`.

Um simples exemplo do *for* com o `range` é dado abaixo. Neste caso somente os números de 1 a 9 são mostrados. Já no código 4.7 exemplificamos a lista de elementos de 1 a 10 com incremento 2.

Código 4.6: Exemplo de *loop for* de incremento 1

```

1 #Programa que escreve os inteiros no intervalo [1, 10[
2 for i in range(1, 10):
3     print(i)

```

Código 4.7: Exemplo de *loop for* com incremento diferente de 1

```

1 #Programa que escreve os inteiros no intervalo [1, 11[ de 2 em 2
2 for i in range(1, 11, 2):
3     print(i)

```

Para finalizar apresentamos com os códigos 4.8 e 4.9 duas formas simples usando *loop* de se obter a soma dos 10 primeiros termos de uma PA, com primeiro elemento 2 e incremento 2; a primeira com o *loop while* e o segundo com o *loop for*.

Código 4.8: Exemplo do *while* para a soma dos 10 primeiros naturais pares

```

1 #Programa que soma os inteiros na PA de razão 2, no intervalo [2,20]
2 x, soma = 1, 0
3 while x <= 10:
4     y = 2 * x
5     print(y)
6     x += 1
7     soma = soma + y
8 print(soma)

```

Incluímos o acumulador *soma* e atribuímos o valor 0, por ser elemento neutro da soma. Escrevemos ainda *soma = soma + y* dentro da indentação, para que somemos todos os elementos de *y*.

Na estrutura *for* precisamos definir o intervalo, pois queremos um *loop* com os 10 primeiros naturais pares. Por isso o *range* deve ficar ajustado no intervalo [1, 11[. Evidentemente que o resultado para essa soma poderia ser construindo-se o *for* com *range(2, 21, 2)* e a variável *soma* ser acumulativa sobre o contador *i* (*soma += i*)⁶.

Código 4.9: Exemplo do *for* na Soma de uma PA

```

1 #Programa que soma os inteiros na PA de razão 2, no intervalo [1, 11]
2 soma=0
3 for i in range(1, 11):
4     y=2*i
5     print(y)
6     soma=soma+y
7 print(soma)

```

⁶Procedimento com pensamento análogo poderia ser usado no *loop while*!

4.4 Bibliotecas

Como já citado anteriormente, o *Python* é uma linguagem de programação de código aberto, o que significa que inúmeros programadores podem contribuir com soluções para problemas que originalmente o programa não resolveria. Pela popularidade do *Python*, existem uma quantidade considerável de bibliotecas que podem solucionar muitos problemas de diferentes naturezas.

Para Yamamoto (2022), dentre as diversas linguagens de programação, o *Python* se destaca por oportunizar um alto nível; fácil acesso e construção de códigos pelo seus inúmeros termos oriundos da língua inglesa. Ainda segundo o autor, os principais em voga usos da ferramenta são: *machine learning*, mercado financeiro, *Data Science*, desenvolvimento *web* e automação.

No mercado financeiro, o *Python* tem sido utilizado por diversos motivos, entre eles a importação de dados das bolsas de valores, como por exemplo, a B3; o tratamento de dados, como a organização em linhas e colunas; a agilidade na realização dos cálculos matemáticos; e a organização dos dados e resultados obtidos em gráficos intermediada por suas bibliotecas; conforme percebemos em diversos trabalhos, como os de Yamamoto (2022) e Menezes (2019).

As bibliotecas utilizadas, que achamos mais compatíveis com nosso trabalho, são: *yfinance*; *pandas*; *numpy*; *scipy.optimize*; *matplotlib* e o *plotly*. Para evocar qualquer dessas bibliotecas, basta usar o comando `import`.

4.4.1 *yfinance*

A biblioteca *yfinance* importa para o programa os dados das bolsas de valores B3, cotações de moedas fiduciárias, e de diversas criptomoedas coletados do site [YAHOO FINANCE](#).

4.4.2 *pandas*

Escolhemos o *pandas* porque é uma das bibliotecas mais conhecidas e poderosas do *Python* para organizar, tratar e manipular dados. Quando queremos organizar os dados em uma tabela ou renomear a própria tabela ou uma linha ou coluna, faremos isso por intermédio do `DataFrame`, que é uma tabela utilizada como recurso do *pandas*.

O *Pandas* é a biblioteca *Python* para manipulação e análise de dados. Permite ler, manipular, agregar e plotar dados com suas mais variadas funções. A biblioteca é essencial para o tratamento e limpeza de dados de maneira eficiente. (NEVES; COSTA, 2022, p. 67)

Por meio do *Pandas* podemos fazer cálculos estatísticos, como os de média aritmética;

moda; desvio padrão; covariância; e coeficiente de correlação de maneira mais ágil, por apenas um comando. Sua característica em organizar seus dados em listas, acaba por facilitar também os cálculos sobre matrizes.

4.4.3 *Numpy*

A biblioteca *numpy*, em combinação com a biblioteca *pandas*, são responsáveis por boa parte dos cálculos deste trabalho. “O Numpy é a biblioteca *Python* para trabalhar com a computação numérica para a utilização de inúmeros cálculos matemáticos” (NEVES; COSTA, 2022, p. 67).

4.4.4 *scipy.optimize*

“A biblioteca *scipy.optimize* é popular na programação científica. Direcionada para modelos matemáticos, foi desenvolvida para trabalhar com arrays *Numpy*.” (NEVES; COSTA, 2022, p. 67) Completando as bibliotecas utilizadas neste trabalho com a finalidade de executar cálculos matemáticos. No entanto, ela servirá apenas para comparação dos nossos resultados.

A organização visual dos dados e resultados obtidos neste trabalho, foram deixados aos cuidados das bibliotecas: *Matplotlib* e *Plotly*

4.4.5 *Matplotlib*

Foi escolhida pelos inúmeros recursos gráficos,

“O *Matplotlib* é a biblioteca mais famosa para a visualização de dados em forma de gráficos. Deve-se atentar que a *Matplotlib* é uma extensão da biblioteca *Pandas* e *Numpy*. Para usá-la é necessário ter as duas bibliotecas instaladas e inseridas no programa. Desta forma, a *Matplotlib* possui funções para construir gráficos de pizza, de barras, de linha e histograma utilizando um conjunto de dados de caráter unidimensional ou bidimensional.” (YAMAMOTO, 2022, p. 25)

4.4.6 *Plotly*

A biblioteca “Plotly é uma plataforma colaborativa de gráficos e análises baseada na *web*. Permite a criação de gráficos interativos com fácil implementação [Inc. 2015]”. (MARTINS; LENHARD, 2021, s.p.).

Na próxima seção, aprofundaremos os conceitos até aqui apresentados, com a esperança de fomentar o uso da estatística intermediada pelo Modelo de Markowitz, usando o *Python*.

Avaliação de Carteira

Neste capítulo apresentamos uma construção aproximada da Fronteira de Markowitz, e sobre ela calculamos o máximo índice de Sharpe para obter o ponto de melhor risco \times retorno, também aproximado, que retorna os pesos ‘adequados’ para constituir os investimentos de ativos na carteira de renda variável. Até chegar esse resultado, todos os passos, importação e tratamento de dados, criação de variáveis e análises têm discussão sobre a confecção dos códigos confeccionados no *Python*. Ao final os rendimentos da carteira hipotética é comparada com os rendimentos em que os pesos foram resultantes de otimização nos elementos do modelo de Markowitz. Todos os comandos do desenvolvimento estão hospedados no *notebook* do *Colaboratory*: [Carteiras Ótimas – Hernandes Macedo](#), para acompanhamento de qualquer pessoa interessada, com possibilidade de fazer cópia.

5.1 Composição da carteira

Pensamos numa simulação para realização do estudo de composição de carteira utilizando-se uma carteira hipotética, distribuída como se apresenta da Tabela 5.1. Não há aqui indicação de composição de uma carteira, e somente se destina a uma exemplificação de aplicação. Os percentuais de cada aplicação estão na Tabela 5.2, junto os nomes ou *tickers* e *tokens* de cada ativo.

Tabela 5.1: Distribuição da carteira por categoria

Categoria do Ativo	Peso
Renda Fixa	7,5%
Ações	60%
Fundos Imobiliários	15%
Fundos Investimento no Exterior	12,5%
Criptomoedas	5,0%

Fonte: do autor.

Tabela 5.2: Distribuição dos ativos.

	Ativo	Peso:
Renda Fixa:	Tesouro Selic 2026 (LFT)	2,5%
	Tesouro Prefixado 2026 (LTN)	2,5%
	Tesouro IPCA+ 2029 (NTN-B Principal)	2,5%
Ações:	BBBS3	7,5%
	EQTL3	7,5%
	SBSP3	7,5%
	MULT3	7,5%
	GGBR4	7,5%
	PETR4	7,5%
	BEEF3	7,5%
	EQPA3	7,5%
Fundos Imob.:	ALZR11	7,5%
	CVBI11	7,5%
Fundos Investimento no Exterior:	NASD11	7,5%
	IVVB11	5,0%
Criptomoedas:	BTC (Bitcoin)	3,0%
	ETH (Ethereum)	1,0%
	SOL (Solana)	1,0%

Fonte: do autor.

Acreditamos que uma composição de carteira seja uma boa discussão para o professor realizar em sala de aula por conta que ele poderá citar vários fatores que podem alterar o rendimento total de um grupo ou mais de ativos. E que ter isso em mente traz alertas de que investir não necessariamente tem a atitude de colocar seu dinheiro num ramo de capitalização e achar não precisará alterar a configuração da carteira. O investidor deve acompanhar os acontecimentos nacionais, ou mesmo globais a fim de salvaguardar seu capital ou almejar maior rentabilidade.

Apesar de não ser necessário listar os ativos da carteira de um investimento em planilha, tal documento serve, dependendo da quantidade de ativos, para facilitação de distribuição do professor a seus alunos, assim como uma visualização rápida dos ativos. Esse arquivo pode ser construído em software livre, como o **calc** do **Libre Office**, com a extensão **.ods**, e posteriormente criado um **.csv**¹ por *Salvar como*. É preferível o arquivo em **.csv** porque é um tipo de arquivo que independe de sistema operacional ou software de gerenciamento de planilha.

Para a carteira da Tabela 5.2 o arquivo **ativos.csv** tem seu conteúdo como na Figura 5.1 abaixo!

¹*comma separated values* = valores separados por vírgula

Figura 5.1: Carteira criada em planilha e salva como `ativos.csv`

	A	B	C
1	Ativos	Categoria	Valor investido (em reais)
2	Tes. Selic 2026	Renda Fixa	250
3	Tes. Pre. 2026	Renda Fixa	250
4	Tes. IPCA+ 2029	Renda Fixa	250
5	BBSE3	Ação	750
6	EQTL3	Ação	750
7	SBSP3	Ação	750
8	MULT3	Ação	750
9	GGBR4	Ação	750
10	PETR4	Ação	750
11	BEEF3	Ação	750
12	EQPA3	Ação	750
13	ALZR11	Fundo Imobiliário	750
14	CVBI11	Fundo Imobiliário	750
15	NASD11	Fundo Inv. No Exterior	750
16	IVVB11	Fundo Inv. No Exterior	500
17	BTC	Criptomoeda	300
18	ETH	Criptomoeda	100
19	SOL	Criptomoeda	100

Fonte: do autor

Assim como a composição da carteira, os valores e pesos, conforme a distribuição por categoria, também foram distribuídos aleatoriamente. Em renda fixa foi somente destinado 7,5% (nada aconselhável em se pensando risco mínimo). Em renda variável, foram os exagerados 92,5% do total do investimento, ou seja, do valor total de R\$ 10.000,00, foi simulado um investimento com R\$ 9.250,00 para renda variável. Apesar de não termos tomado ponderações comuns no mundo dos investimentos, aconselhamos aos professores discutir a distribuição de ativos na carteira. Por exemplo, o Tesouro Selic (LFT) (retorno pós-fixado) tem liquidez diária, podendo ser então um investimento resgatado no mesmo dia ou no dia seguinte ao pedido, dependendo horário, e sem perdas, configurando-se portanto como parte de uma reserva de emergência.

5.2 Importação e tratamentos de dados dos ativos

5.2.1 Importação da carteira

Após ter o arquivo com as informações dos ativos, devemos carregá-lo no *Python* para manipulação. Dentre algumas bibliotecas disponíveis para tratamento de dados, decidimos usar o *Pandas*, por ser a mais conhecida e de exigir relativamente de pequeno grupo de comandos para as necessidades que aqui nos propomos. O *Pandas* foi criado em 2008 por Wes McKinney, e tem funções para análise, limpeza, exploração e manipulação de dados. O nome refere-se a “Panel Data” e “Python Data Analysis”. Tem grande uso porque tem recursos que permitem analisar e fazer conclusões sobre conjuntos de dados enormes através de teoria estatística. Ela por exemplo retorna informações de valores mínimo, máximo e médio de uma coluna, ou mesmo se existe uma correlação de valores entre elas.

Para carregar o arquivo `ativos.csv` pelo *Pandas*, primeiro devemos importar essa biblioteca, e em seguida usar um comando específico para ler o arquivo. No Código 5.1 abaixo se tem os comandos para isso. Neste caso a variável `df` foi utilizada para armazenamento da tabela, que no *Pandas* tem a denominação de ‘Data Frame’². Como resultado se tem o que se mostra na Figura 5.2.

Código 5.1: Importando o *Pandas* e leitura dos ativos

```
1 import pandas as pd          # importação do pandas
2 df = pd.read_csv('ativos.csv') # carregamento de arquivo .csv
3 df
```

Figura 5.2: Data Frame carregado com a carteira pelo *Pandas*

	Ativos	Categoria	Valor investido (em reais)
0	Tes. Selic 2026	Renda Fixa	250
1	Tes. Pre. 2026	Renda Fixa	250
2	Tes. IPCA+ 2029	Renda Fixa	250
3	BBSE3	Ação	750
4	EQTL3	Ação	750
5	SBSP3	Ação	750
6	MULT3	Ação	750
7	GGBR4	Ação	750
8	PETR4	Ação	750
9	BEEF3	Ação	750
10	EQPA3	Ação	750
11	ALZR11	Fundo Imobiliário	750
12	CVBI11	Fundo Imobiliário	750
13	NASD11	Fundo Inv. No Exterior	750
14	IVVB11	Fundo Inv. No Exterior	500
15	BTC	Criptomoeda	300
16	ETH	Criptomoeda	100
17	SOL	Criptomoeda	100

Fonte: do autor

Reparemos que o cabeçalho da tabela aparece, e que os índices das linhas também, iniciados por 0, que é o padrão usado pelo *Python*. Devemos interpretar um ‘Data Frame’ como um *array* bidimensional, em que pode se endereçar qualquer elemento informando a linha e a coluna onde está na tabela. Neste caso do *Pandas* o atributo `loc` deve ser usado,

²Quando um ‘data frame’ tem um número grande de linhas, a ação do `print` sobre a variável só fará a apresentação das cinco primeiras e das cinco últimas linhas. Para ver todas é preciso usar, por exemplo, `print(df.to_string())`

e, por exemplo, para referência do *ticker* **PETRA** basta usar `df.loc[8][0]`. Mas, diferentemente de um *array* do tipo lista habitual no *Python* não se pode referenciar uma só coluna ou uma só linha por `:`, já que o ‘Data Frame’ é semelhante um dicionário, e por isso, o comando `df.loc[[0,2]]` retornaria as linhas de índices 0 e 2 junto com o cabeçalho da tabela `df` e à esquerda os respectivos índices dessas linhas. Sendo assim, a soma de todos os valores investidos, armazenados na última coluna, deve ter uma sintaxe diferente. Para isso devemos informar o cabeçalho da coluna da tabela!

Cada coluna de um ‘DataFrame’ deve ser referenciado pelo cabeçalho. Por exemplo, se desejarmos calcular o valor mínimo, o máximo, o total, a média, a mediana e a moda dos valores investidos, poderemos fazer o que se mostra no Código 5.2 a seguir:

Código 5.2: Exemplo de cálculos de mínimo, máximo, somatória, média, mediana e moda de uma coluna específica de um ‘Data Frame’ do *Pandas*

```

4 valores = df["Valor Investido (em reais)"] # uma coluna em específico
5 minim_val = valores.min() # mínimo valor na coluna
6 maxim_val = valores.max() # máximo valor na coluna
7 total_val = valores.sum() # somatória dos valores
8 media_val = valores.mean() # média aritmética
9 medianval = valores.median() # mediana
10 moda_val = valores.mode() # moda
11 print(minim_val, maxim_val, total_val, media_val, medianval, moda_val)

```

Com exceção da mediana, moda e valor total encontrado acima, o *Pandas* também traz informações estatísticas básicas com um simples comando. Usando o comando `df.describe()` teremos de nossos dados as saídas, em sequência, de quantidade total de linhas com valor numérico não nulo, média, desvio-padrão, valor mínimo, primeiro quartil (percentil 25), segundo quartil, terceiro quartil e valor máximo.

Figura 5.3: Resultado do comando `df.describe()`.

Valor investido (em reais)	
count	18.000000
mean	555.555556
std	263.399098
min	100.000000
25%	262.500000
50%	750.000000
75%	750.000000
max	750.000000

Fonte: do autor

Um recurso interessante que o *Pandas* possui é o de poder fazer operações associadas a grupos de dados. O comando `groupby()` permite em nosso caso agrupar as categorias de ativos e fazer operações sobre cada uma. Poderemos por exemplo verificar qual valor investido em cada categoria está sendo feito. O Código 5.3 transcreve essa ideia! Nesse caso é necessário ser explicitada que coluna se deve agrupar as variáveis, que foi aqui '`Categoria`'. Por padrão, o *Pandas* apresenta o agrupamento em ordem alfabética de variáveis, daí se usou o argumento `sort = False` para permanecer com a mesma sequência que se apresenta na tabela. Por fim `.sum()` foi usado para indicar que os valores numéricos associados às variáveis '`Categoria`', de qualquer outra coluna, devem ser somados. O resultado se apresenta na Figura 5.4.

Código 5.3: Agrupamento de tipos de investimento com `groupby()`

```
12 somacateg = df.groupby('Categoria', sort = False).sum()
13 print(somacateg)
```

Figura 5.4: Soma dos investimentos agrupados em categoria.

Valor investido (em reais)	
Categoria	
Renda Fixa	750
Ação	6000
Fundo Imobiliário	1500
Fundo Inv. No Exterior	1250
Criptomoeda	500

Fonte: do autor

Para obter informações de ativos em tempo passado existem pelo menos dois *sites* bastante usados. Um é o [Google Finance](#) e o outro o [Yahoo Finance](#). E ambos possuem pacotes no *Python* possibilitando baixar as informações para os códigos (mas havendo conexão com a internet!). Com o *googlefinance* é possível obter informações quase que em tempo real, mas o *yfinance* tem mais recursos sobre histórico de ativos, por isso vamos usar o *yfinance*, mesmo que, ambos, não possuam acesso as ativos do Tesouro Direto. Nesse caso usaremos um outro caminho.

5.2.2 Busca de informações do Tesouro

Existem duas maneiras para obtenção de informações do Tesouro Direto via arquivo. Mas, até a data do desenvolvimento deste trabalho, infelizmente tais arquivos não são equivalentes.

Eles diferem nas datas de histórico de cotações, problema que foi percebido somente ao experimentar as simulações. Por esse motivo, vamos mostrar as duas formas de coleta dos arquivos.

Na primeira forma devemos acessar [Tesouro Nacional Transparente](#), escolher na aba **Categoria** a opção **Dados Abertos**, e nessa nova página clicar em **Taxas de Títulos Ofertados pelo Tesouro Direto**. Em seguida baixar o arquivo [PrecoTaxaTesouroDireto.csv](#). Esse mesmo arquivo pode ser obtido diretamente pelo código em *Python*, criando uma variável no editor com o endereço completo do arquivo³. Para isso, clicando-se em ⓘ (Mais informações), abrirá uma nova janela com o endereço do arquivo ao lado de URL:. Copie-o para a variável destinada no editor. Essa ação está simulada no Código 5.4 abaixo⁴, efetuada no dia 17/11/2023.

Código 5.4: Carregamento de dados do Tesouro Nacional, via link (em 17/11/2023)

```

14 link = 'https://www.tesourotransparente.gov.br/ckan/dataset/' +
15       'df56aa42-484a-4a59-8184-7676580c81e3/resource/' +
16       '796d2059-14e9-44e3-80c9-2d9e30b405c1/download/' +
17       'PrecoTaxaTesouroDireto.csv'
18 tesouro = pd.read_csv(link, sep = ';', decimal = ',')
19 print(tesouro)

```

Aqui é importante saber que um dado é separado do outro por ponto-e-vírgula; e que a separação de uma coluna para outra corresponde simplesmente a separação de um valor monetário inteiro da sua parte decimal! Dessa forma, para o carregamento do arquivo foi preciso especificar na leitura o separador de colunas e o símbolo decimal adotado (linha 18). A Figura 5.5 apresenta o que se obtém com a leitura do arquivo⁵.

Ao consultar o arquivo do tesouro nacional, podemos ver que existem 6 colunas, mas na primeira linha e primeira coluna são colocadas informações, que são referentes a (ver [Entenda tabela de preços e taxas](#)):

Título: apresenta os títulos públicos atualmente negociados no Tesouro Direto;

Vencimento: corresponde à data de vencimento do título, que representa a data em que o título é resgatado automaticamente pelo sistema. Nesta data, o título deixa de existir

³Dependendo da sua conexão com a internet, usar dessa segunda estratégia pode demandar mais tempo que o caso de já se ter o arquivo `.csv` baixado.

⁴Reparemos que por conta do `link` ser grande, houve a necessidade de cortá-lo (com `\`) em vários pedaços, e que foram concatenados na sintaxe do *Python* com `+`

⁵Para verificar se o que foi dito acima quanto à distribuição das informações entre ponto-e-vírgula e colunas separadas faz sentido, use `.loc`; como por exemplo:

```
print(tesouro.loc[0][0], tesouro.loc[0][1], tesouro.loc[0][2], tesouro.loc[0][3],
      tesouro.loc[0][4], tesouro.loc[0][5], tesouro.loc[0][6], tesouro.loc[0][7])
```

Figura 5.5: Arquivo carregado do tesouro direto

	Tipo Titulo	Data Vencimento	Data Base	Taxa Compra	Manha	Taxa Venda	Manha	PU Compra	Manha	PU Venda	Manha	PU Base	Manha
0	Tesouro IPCA+ com Juros Semestrais	15/08/2012	01/04/2010		6.22		6.26		1909.76		1908.15		1906.47
1	Tesouro IPCA+ com Juros Semestrais	15/05/2013	01/04/2010		6.35		6.39		1930.36		1928.32		1926.60
2	Tesouro IPCA+ com Juros Semestrais	15/05/2011	01/04/2010		5.39		5.41		1959.70		1959.30		1957.63
3	Tesouro IPCA+ com Juros Semestrais	15/08/2010	01/04/2010		4.87		4.89		1926.22		1926.08		1924.48
4	Tesouro IPCA+	15/05/2015	01/04/2010		6.36		6.42		1389.68		1385.68		1384.44
...
134539	Tesouro IPCA+ com Juros Semestrais	15/08/2006	28/02/2005		10.59		10.61		1415.30		1414.93		1414.06
134540	Tesouro IPCA+ com Juros Semestrais	15/05/2009	28/02/2005		9.26		9.30		1364.96		1363.13		1362.35
134541	Tesouro IPCA+ com Juros Semestrais	15/05/2045	28/02/2005		9.05		9.15		1045.57		1034.59		1034.01
134542	Tesouro IPCA+ com Juros Semestrais	15/08/2024	28/02/2005		8.83		8.91		1123.55		1114.95		1114.33
134543	Tesouro IPCA+ com Juros Semestrais	15/05/2015	28/02/2005		8.83		8.91		1253.33		1246.60		1245.91

134544 rows x 8 columns

Fonte: do autor

e o Tesouro Nacional repassa o valor equivalente ao resgate ao Agente de Custódia do investidor, que posteriormente repassa ao investidor, após a dedução de impostos e taxas;

Taxa compra (a.a.): corresponde à rentabilidade bruta ao ano que o investidor receberá caso adquira o título naquele momento e o mantenha até sua data de vencimento. No caso dos Tesouro IPCA+ (NTN-B Principal), Tesouro IPCA+ com Juros Semestrais (NTN-B) e Tesouro Selic (LFT), além da rentabilidade mostrada, o investidor receberá também a variação do indexador de cada título (IPCA, IPCA, e Selic, respectivamente) entre a data da compra e a data de vencimento. A taxa de compra é garantida se o investidor ficar com o título até sua data de vencimento. Caso opte pela venda antecipada do seu título, o investidor receberá o valor de mercado do mesmo;

Taxa venda (a.a.): refere-se à taxa bruta pela qual o investidor poderá vender seu título ao Tesouro Nacional antes de sua data de vencimento. A coluna referente a essa taxa é preenchida todos os dias, já que o Tesouro Nacional realiza a recompra de títulos vendidos via Tesouro Direto diariamente. Note que, no caso da venda antecipada, o Tesouro Nacional recompra o título com base em seu valor de mercado. Caso o investidor mantenha seu título até o vencimento, receberá a rentabilidade pactuada no momento da compra, independentemente das condições de mercado;

PU compra: (Preço unitário – compra) corresponde ao preço que o investidor deve pagar para adquirir uma unidade do título vendido no Tesouro Direto. Não é necessário comprar um título inteiro, sendo possível comprar frações do título. A quantidade mínima de compra é 0,01 do título (1%). O investidor pode comprar múltiplos de 0,01, como 0,04 título; 0,05 título; 0,08 título e assim por diante. Não é possível comprar, por

exemplo, 0,015 título ou 5,73 títulos. O valor mostrado na tabela refere-se ao preço de compra um título, e deve ser ajustado para a quantidade desejada para compra pelo investidor;

PU venda: (Preço unitário – compra) corresponde ao valor bruto recebido pelo investidor caso realize a venda de seu título ao Tesouro Nacional antes de sua data de vencimento. O valor mostrado na tabela refere-se ao preço de um título, e deve ser ajustado para a quantidade desejada para venda pelo investidor. Por exemplo, no caso da venda de 0,1 título, o preço de venda será 10% do valor mostrado. Essa coluna é preenchida todos os dias, já que o Tesouro Nacional realiza a recompra de títulos vendidos via Tesouro Direto diariamente. Note que, no caso da venda antecipada, o Tesouro Nacional recompra o título com base em seu valor de mercado. Caso o investidor mantenha seu título até o vencimento, receberá a rentabilidade pactuada no momento da compra, independentemente das condições de mercado.

Além do **Título**, **Data Vencimento**, e **Data Base**, na tabela são informadas uma taxa de compra, taxa de venda (horário da manhã), PU compra, PU venda (horário da manhã) e PU base manhã, correspondendo 8 tipos de dados distribuídos em 6 colunas.

O próximo passo para tratamento das informações é extrair do arquivo do tesouro direto as cotações dos ativos de nossa carteira. Nessa tarefa basta-nos usar o comando `.loc` novamente. Ele agora fará a busca dentro do ‘Data Frame’ as linhas que contenham uma especificação (ou especificações) que caracterizem cada ativo nosso.

No site [Títulos do Tesouro Direto](#) temos informações de todos os títulos abertos. Vamos utilizar para busca das linhas do arquivo as datas de vencimento informadas aí! O Código 5.5 que criará novos ‘Data Frame’ dos ativos está abaixo, onde podemos ver que o argumento dentro do `.loc` é justamente a coluna do ‘Data Frame’ com a data de vencimento. Cada um tem conhecidos números de linhas e colunas com o comando `.shape` após o nome da variável, como também está no código.

Código 5.5: Criação do ‘Data Frame’ dos ativos de renda fixa da carteira (arquivo via link)

```
20 tesourosel2026 = tesouro.loc[tesouro['Data Vencimento'] == '01/03/2026']
21 tesouropre2026 = tesouro.loc[tesouro['Data Vencimento'] == '01/01/2026']
22 tesouroipc2029 = tesouro.loc[tesouro['Data Vencimento'] == '15/05/2029']
23 print(tesourosel2026.shape, tesouropre2026.shape, tesouroipc2029.shape)
```

Esses novos ‘Data Frame’s têm no entanto os mesmo índices das linhas do `tesouro`. Isto significa que, por exemplo, o primeiro índice `tesouropre2026` é o mesmo da enumeração da primeira ocorrência do ativo Tesouro Prefixado 2026 dentro do ‘Data Frame’ `tesouro`.

Para fazer com que os índices comecem em 0, o comando `.reset_index(drop = True)` deve ser aplicado em cada ‘Data Frame’ novo, como mostra o Código abaixo:

Código 5.6: Renumeração dos índices dos novos ‘Data Frame’ criados a partir do arquivo via link

```
24 tesourosel2026 = tesourosel2026.reset_index(drop = True)
25 tesouropre2026 = tesouropre2026.reset_index(drop = True)
26 tesouroipc2029 = tesouroipc2029.reset_index(drop = True)
```

O padrão de datas no *Python* segue o formato `aaaa-mm-dd`, e no entanto as datas do tesouro são dadas em `dd/mm/aaaa`. E já que existe a possibilidade de comparar os investimentos de diversas categorias, com algumas delas obtidos históricos do *Yahoo Finance*, iremos converter para as datas no formato que o *Python* usualmente trabalha. O *Pandas* faz essa conversão usando simplesmente a função `to_datetime` sobre os dados, que no Código 5.7 abaixo foi aplicado à coluna ‘Data Base’, que informa o valor do ativo em determinado tempo antes do vencimento. O argumento `dayfirst = True` faz exatamente a transformação das datas `dd/mm/aaaa` para o formato `aaaa-mm-dd`.

Código 5.7: Conversão de datas do tesouro direto para o formato `aaaa-mm-dd` (arquivo via link)

```
27 tesourosel2026['Data Base']=pd.to_datetime(tesourosel2026['Data Base'],
    dayfirst=True)
28 tesouropre2026['Data Base']=pd.to_datetime(tesouropre2026['Data Base'],
    dayfirst=True)
29 tesouroipc2029['Data Base']=pd.to_datetime(tesouroipc2029['Data Base'],
    dayfirst=True)
```

Se em algum momento houver a necessidade criar um único ‘Data Frame’ com os ativos do tesouro e os demais ativos baixados do *Yahoo Finance*, por exemplo; as datas é que serviram como balizadoras nessa criação. E sendo assim, a coluna de data deverá ser renomeada, ou do tesouro, ou a do outro ‘Data Frame’. O Código 5.8 abaixo mostra a realização dessa tarefa sobre os dados do tesouro, usando o comando `.rename`, que neste caso foi aplicado também à coluna ‘PU Base Manha’ para a designação do tipo ativo do tesouro. Atente-se que a renomeação aconteceu para um novo ‘Data Frame’!

Código 5.8: Renomeação de datas e valores dos ativos do tesouro

```
30 tessel2026=tesourosel2026.rename(columns={'Data Base': 'Date', 'PU Base Manha':
    'Tes. Selic 2026'})
31 tespre2026=tesouropre2026.rename(columns={'Data Base': 'Date', 'PU Base Manha':
    'Tes. Pre. 2026'})
32 tesipc2029=tesouroipc2029.rename(columns={'Data Base': 'Date', 'PU Base Manha':
    'Tes. IPCA+ 2029'})
```

A segunda forma de obtenção dos arquivos com as cotações dos títulos do Tesouro Direto é baixando em separado o histórico, mas por ano, de cada título. O endereço de hospedagem nessa modalidade está em [Histórico de Pregos e Taxas](#). Escolhemos o ano de 2022, e assim os arquivos *.xls* baixados foram LFT_2022 (Tesouro Selic), LTN_2022 (Tesouro pré-fixado) e NTN-B_Principal_2022 (Tesouro IPCA+). Neste caso cada um dos arquivos de títulos estão no formato do *EXCEL*, e conseqüentemente um outro comando de leitura do *Pandas* é usado. O Código 5.9 abaixo substitui os Códigos 5.4, 5.5, 5.6, 5.7 e 5.8.

Código 5.9: Leitura e tratamento de histórico de títulos do tesouro (arquivos por ano)

```

20 tesourosel2026 = pd.read_excel('LFT_2022.xls', header = 1)
21 tesouropre2026 = pd.read_excel('LTN_2022.xls', header = 1)
22 tesouroipc2029 = pd.read_excel('NTN-B_Principal_2022.xls', header = 1)
23 tesourosel2026['Dia'] = pd.to_datetime(tesourosel2026['Dia'], dayfirst=True)
24 tesouropre2026['Dia'] = pd.to_datetime(tesouropre2026['Dia'], dayfirst=True)
25 tesouroipc2029['Dia'] = pd.to_datetime(tesouroipc2029['Dia'], dayfirst=True)
26 tessel2026=tesourosel2026.rename(columns={'Dia': 'Date', 'PU Base Manhã': 'Tes.
    Selic 2026'})
27 tspre2026=tesouropre2026.rename(columns={'Dia': 'Date', 'PU Base Manhã': 'Tes.
    Pre. 2026'})
28 tesipc2029=tesouroipc2029.rename(columns={'Dia': 'Date', 'PU Base Manhã': 'Tes.
    IPCA+ 2029'})

```

Nas linhas 20, 21 e 22 desse Código 5.9 estão os comandos do *Pandas* para leitura dos arquivos do *Excel*. O argumento `header = 1` (cabecalho começando da linha de índice 1, e não 0) foi usado porque os arquivos do tesouro nesse formato tem sua primeira linha informação de data de vencimento do título, o qual não nos serve para uso dos dados. Diferentemente dos arquivos *.csv* tratados anteriormente, as data das cotações não estão na designação 'Data Base' e sim 'Dia'; por isso as linhas 23 a 25 fazem o modificação de formato de data conforme o cabecalho desses novos arquivos. As linhas 26, 27 e 28, análogas às do código 5.8, são usadas somente na necessidade posterior de agrupar todos os ativos da carteira!

Repare que as linhas de identificação do Código 5.9 reiniciam a partir da linha 20; mas, os demais códigos que serão apresentados seguirão a sequência de enumeração anterior (linha 32) porque as simulações ocorreram sobre a primeira forma de obtenção dos arquivos do tesouro.

Os investimento em renda fixa aqui colocados foram com objetivo de, além de mostrar seus desempenhos, servirem de variável no cálculo do *Risk Free* citado no capítulo 3. Então convém criar um 'Data Frame' para os dados de renda fixa e dele extrair um retorno médio de aplicação num determinado período.

O Código 5.10 abaixo mostra o procedimento de concatenação dos dados do tesouro

da carteira, a fim de posteriormente fazer cálculos em datas específicas.

Código 5.10: Concatenação de dados do tesouro

```

33 fixa=pd.DataFrame( tessel2026 [[ 'Date', 'Tes. Selic 2026' ]])
34 fixa=fixa.merge( tespre2026 [[ 'Date', 'Tes. Pre. 2026' ]], on='Date',how='inner')
35 fixa=fixa.merge( tesipc2029 [[ 'Date', 'Tes. IPCA+ 2029' ]], on='Date',how='inner')
36 fixa=fixa.set_index( 'Date')
37 fixa=fixa.sort_index()
38 print( fixa.head())

```

Na linha 33 um novo ‘Data Frame’, `fixa`, é criado, com somente as colunas `Date` e `Tes. Selic 2026` do `tesourosel2026`.

Nas linhas 34 e 35 houverem as concatenações ao `fixa`, através do comando `.merge`. Notemos que com o `.merge` a junção dos ‘Data Frame’ ocorreram acrescentando-se um título por vez, e em cada uso houve a especificação de que colunas gostaríamos de inserir, mas, mesclando uma coluna de referência na junção, no caso a coluna ‘Date’ atribuída com o argumento `on = ‘Date’`. Para as datas compreenderem as de menor período, houve a necessidade de dizer que a mesclagem dos dados fosse restringindo-se à menor. Isso é atribuído à entrada `how = ‘inner’`, que é a padrão.

Com o comando `.merge` o novo ‘Data Frame’ terá seus índices inteiros, por esse motivo usamos o comando `.set_index`, na linha 36, o qual redefiniu os índices das linhas para ‘Date’. Ao observamos todos os dados do tesouro, vemos que eles se apresentam em ordem decrescente de datas, daí usamos o comando `.sort_index()` (linha 37) para reordená-los a partir da menor.

De posse dos dados, falta-nos definir que período de investigação serão tratados os ativos. Isso é simplesmente informado por variáveis, que aqui decidimos definindo para o intervalo de 01/01/2023 a 17/11/2023, cujos inputs estão em 5.11 abaixo. Na linha 40 se tem, após o comentário, o comando `.date.today()`, para opcionalmente definir o dia final a data de execução do código.

Código 5.11: Definição das datas início e fim para download de dados

```

39 diaini = '2023-01-01' # já no formato aaaa - mm - dd
40 diafin = '2023-11-17' # dt.date.today() # <— no dia de execução do código

```

Precisamos observar se as datas definidas como inicial e final para as simulações estão contidas nos dados do tesouro direto! Uma verificação rápida pode ser realizada, em nosso caso, como o comando `print(fixa.info())`. Por ele vemos que somente o dia inicial não está contido nos dados, e que existem 214 registros, a partir do dia 09/01/2023, no período desejado (ver Figura 5.6).

Figura 5.6: Informações do ‘Data Frame’ para o tesouro direto com o comando `.info()`

```

<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 214 entries, 2023-01-09 to 2023-11-16
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Tes. Selic 2026        214 non-null    float64
1   Tes. Pre. 2026         214 non-null    float64
2   Tes. IPCA+ 2029        214 non-null    float64
dtypes: float64(3)
memory usage: 6.7 KB
None

```

Fonte: do autor

5.2.3 Busca de informações no Yahoo Finance

Como dito anteriormente, as informações de muitos ativos serão obtidas a partir do *Yahoo Finance* pela biblioteca `yfinance`. Caso não esteja instalada essa biblioteca, devemos realizá-la pelo comando `pip install yfinance`, seja rodando no *Colab* ou no seu computador. Depois disso basta importá-la para o código com o comando `import yfinance as yf`.

Dependendo do ativo hospedado no site, podemos ter informações de vários anos atrás. Então a biblioteca requer que sejam informadas dadas de início e fim.

Outro argumento que deve ser usado para baixar informações é o código (‘ticker’ ou o ‘token’) do ativo. Nos listados na Bovespa, o *Yahoo Finance* destinou o sufixo ‘.SA’ aos ‘tickers’. Enquanto que para criptoativos, apesar de muito de tais ativos poderem ser transacionados entre eles, o site fornece os valores em dólar americano, e por isso é acrescido à terminação ‘-USD’ no ‘token’ da cripto.

Os ‘tickers’ são retornados já em reais, enquanto os ‘tokens’ em dólar consequentemente vamos ter que fazer conversão de dólar para reais. O código para baixar informações, do preço em reais de 1 dólar, é ‘USDBRL=X’. Vamos aproveitar o histórico desse par de moedas para mostrar o que se tem retornado pelo comando `.download` do *Yahoo Finance*. Isso está no Código 5.12 abaixo, do que resulta no que está na imagem 5.7.

Código 5.12: Tomada de cotações do Real a partir do valor do Dólar

```

41 import yfinance as yf
42 brl = yf.download('USDBRL=X', start = diaini, end = diafin)
43 brl

```


Figura 5.7: Histórico de valores em Real a partir do valor em Dólar

	Open	High	Low	Close	Adj Close	Volume
Date						
2023-01-02	5.285500	5.362100	5.281984	5.285500	5.285500	0
2023-01-03	5.360131	5.422100	5.340700	5.360131	5.360131	0
2023-01-04	5.478500	5.479400	5.425700	5.478500	5.478500	0
2023-01-05	5.430100	5.431100	5.369500	5.430100	5.430100	0
2023-01-06	5.349800	5.360859	5.244500	5.349800	5.349800	0
...
2023-11-10	4.933900	4.949700	4.889803	4.933900	4.933900	0
2023-11-13	4.893379	4.940500	4.893379	4.893379	4.893379	0
2023-11-14	4.906200	4.907800	4.848594	4.906200	4.906200	0
2023-11-15	4.864500	4.866500	4.856525	4.864500	4.864500	0
2023-11-16	4.860900	4.883100	4.836291	4.860900	4.860900	0

Fonte: do autor

A variável retornada com o comando `.download` é um ‘Data Frame’, cujas colunas são:

Open: preço no momento de abertura do mercado;

High: maior valor alcançado no dia;

Low: menor valor atingindo no dia;

Close: valor no momento de fechamento;

Adj Close: preço de fechamento após os ajustes para todos os desdobramentos e distribuições de dividendos quando aplicáveis⁶;

Volume: volume total transacionado do ativo durante o dia;

Vamos levar em conta então os valores da coluna ‘Adj Close’ para nossas simulações. Vemos também que os índices das linhas são ‘Date’, que são as datas dos registros dentro do período especificado no comando. Nem todas as datas foram retornadas no intervalo porque o histórico se apresenta sobre o mercado de câmbio de moedas fiduciárias, que acontece em dias comerciais da semana, excetuando-se os feriados. O mesmo ocorre com as informações

⁶Mais informações em [O que é fechamento ajustado?](#)

obtidas para os ativos da Bovespa, mas não nos criptoativos, porque são negociados em todos dias da semana. Por esse fato deveremos fazer adequações para observar o comportamento dos ativos da carteira; ou ignoramos as datas de preços dos criptoativos onde não se tem transações de câmbio e ações da Bovespa, ou; expandimos as datas deste últimos para todo o intervalo de datas que desejarmos analisar! Escolhemos a primeira opção!

Como já temos o ‘Data Frame’ `brl` com as cotações de real frente o dólar, a primeira tarefa é de se ter valores do Real em qualquer dia de transação de criptoativos. Isto pode ser feito mediante os seguintes passos:

- i) Criamos um ‘Data Frame’ de uma criptomoeda qualquer, por exemplo `btc`, com `.download`;
- ii) Renomeamos as colunas ‘Adj Close’ de `btc` e `brl` para, por exemplo ‘BTC’ e ‘Real’;
- iii) Criamos um ‘Data Frame’ `btcbrl` sendo a concatenação das colunas ‘BTC’ e ‘Real’ de `btc` e `brl`;
- iv) Como na coluna ‘Real’ de `btcbrl` não existirão valores nas datas fora dos dias de câmbio, poderíamos usar um comando de preenchimento. No entanto, pela escolha de tomar datas com transações de câmbio, esse passo não seria necessário; mas também, usá-lo não altera a estrutura dos dados neste caso.

No Código 5.13 a seguir temos os passos descritos acima.

Código 5.13: Inclusão de cotações em Real em todas datas do intervalo

```

44 btc = yf.download('BTC-USD', start = diaini, end = diafin) # passo (i)
45 btc = btc.rename(columns = {'Adj Close': 'BTC'}) # passo (ii)
46 brl = brl.rename(columns = {'Adj Close': 'Real'}) # passo (ii)
47 btcbrl = pd.concat([btc['BTC'], brl['Real']], axis = 1, join = 'inner') #(iii)
48 btcbrl.fillna(method = 'ffill', inplace = True) # passo (iv)
49 btcbrl.fillna(method = 'bfill', inplace = True) # passo (iv)

```

Nas linhas 45 e 46 o comando `.rename` realiza a renomeação das colunas, que devem ser indicadas como argumentos com a substituição do nome atual pelo nome desejado.

O comando `.concat` faz a concatenação de ‘Data Frame’, linhas ou colunas de ‘Data Frame’. Em nosso caso, linha 47, somente as colunas ‘BTC’ e ‘Real’ foram concatenadas a fim de compor um novo ‘Data Frame’ com todas as datas no período especificado. As colunas são concatenadas para uma ao lado da outra com a especificação `axis = 1` (`axis = 0` faz concatenação na vertical), enquanto o argumento `join = 'inner'` garante que a junção dos dados será nos de índices comuns de ambos ‘Data Frame’, restringindo-se ao menor, isto é;

a coluna ‘BTC’ diminuirá de tamanho até que se encaixe em todas as datas compreendidas nos dados ‘Real’⁷.

Nas linhas 48 e 49 do código se tem a aplicação do comando `.fillna`. Ele preenche valores especificados nas posições em que ocorre ‘NaN’, mediante a declaração de que valores usar, ou de um método. Em nosso caso, primeiro usamos `method = ffill` porque achamos mais adequada a atribuição da cotação de Real em dias de fim de semana ou feriado, com aquele valor do último dia transacionado (`ffill` faz o preenchimento para frente com o último valor conhecido). Mas com essa decisão, se os primeiros dias tabelados já correspondem a dias sem operações de mercado, seria preciso preencher valores a partir de alguma outra referência. Por isso usamos na sequência `.fillna` com o método `bfill`, já que ele insere nessas datas valor que vem logo iniciando a tabela (`bfill` faz o preenchimento para trás com o primeiro valor conhecido em sequência). O argumento `inplace = True` faz com que a tarefa de preenchimento já se configure permanentemente na variável atribuída no processo⁸. Atente-se no entanto que toda essa ideia descrita aqui no uso de `fillna` não tem efeito com uso de `join = ‘inner’`!

A Figura 5.8 apresenta o resultado da execução do Código 5.13. Mas o procedimento realizado foi somente de se ter valores em reais em todo intervalo desejado. Ainda não transformamos, nem armazenamos, as criptomoedas já com os valores em reais.

Figura 5.8: Obtenção da cotação do Real para cada dólar.

	BTC	Real
Date		
2023-01-02	16688.470703	5.285500
2023-01-03	16679.857422	5.360131
2023-01-04	16863.238281	5.478500
2023-01-05	16836.736328	5.430100
2023-01-06	16951.968750	5.349800
...
2023-11-09	36693.125000	4.909300
2023-11-10	37313.968750	4.933900
2023-11-13	36502.355469	4.893379
2023-11-14	35537.640625	4.906200
2023-11-15	37880.582031	4.864500

[228 rows x 2 columns]

Fonte: do autor

⁷O atributo `outer` para `join`, que é o padrão, faz com que a junção dos dados seja estendendo-se aos índices do maior ‘Data Frame’. Nesse caso, aqueles índices do que contém menos dados são preenchido por NaN.

⁸Não nos pareceu conveniente em nosso caso, mas o preenchimento também pode ser com outro valor, como com a média aritmética de todos os valores da coluna!

Vamos então passar à construção de ‘Data Frame’ com os ativos de criptomoedas. Lembrando que o ‘Data Frame’ da nossa carteira foi `df`, e que nele temos as colunas ‘Ativos’, ‘Categoria’ e ‘Valor investido (em reais)’, podemos em um *loop* varer cada linha de `df`, identificar quais delas se tem a categoria ‘Criptomoeda’ e baixar do *Yahoo Finance* somente as informações ‘Adj Close’. E já junto com o download, multiplicar o valor em dólar ao correspondente em reais da coluna ‘Real’ de `btcb1` encontrado acima. A implementação desse procedimento está descrita no Código 5.14 a seguir.

Código 5.14: Criação de ‘Data Frame’ dos criptoativos

```
50 criptos = pd.DataFrame()
51 for i, token in enumerate(df[ 'Ativos' ]):
52     if df[ 'Categoria' ][i] == 'Criptomoeda':
53         criptos[token] = yf.download( '{}-USD'.format(token),
54             start = diaini, end = diafin)[ 'Adj Close' ] * btcb1[ 'Real' ]
```

Com a linha 50 criamos um ‘Data Frame’ vazio. O *loop for* tem contadores `i` e `token` variando dentro dos índices e dos ‘Ativos’ de `df`, respectivamente. Ou seja, o `enumerate` é uma função que retorna o par índice e elemento do objeto em que ele é aplicado. Por isso tal argumento foi a coluna ‘Ativos’, pois nela é que se apresentam os símbolos dos ativos. Se o índice `i` da coluna ‘Categoria’ é Criptomoeda, far-se-á o download do `token`, mas só nossa coluna de interesse, que também é multiplicada pelo valor do dólar em real em cada data dentro do intervalo de interesse dos dados. No primeiro argumento de `.download` usamos o recurso de inserir a variável `token`, qualquer valor que ela assuma na iteração. A sintaxe `{}-USD` entre apóstrofos, seguida do comando `.format` possibilita realizar essa tarefa. O valor inserido entre as as chaves será o argumento que se tem dentro dos parêntesis de `format`.

Tendo armazenadas as cotações da renda fixa e dos criptoativos, falta-nos fazer o mesmo para os demais ativos. Isso será feito da mesma forma que fizemos com as criptomoe-das, com modificação apenas na condicional para tomar as ações adequadamente, acrescentar o sufixo ‘SA’ ao ‘ticker’ e não ter que multiplicar os valores baixados pelo valor em reais. A transcrição dessa tarefa está no Código 5.15 abaixo.

Código 5.15: Criação de ‘Data Frame’ com tickers da Bovespa

```
55 bolsa = pd.DataFrame()
56 for i, ticker in enumerate(df[ 'Ativos' ]):
57     if df[ 'Categoria' ][i] != 'Renda Fixa' and df[ 'Categoria' ][i] != '
58         Criptomoeda':
59         bolsa[ticker] = yf.download( '{}.SA'.format(ticker),
59             start = diaini, end = diafin)[ 'Adj Close' ]
```

Com isso podemos então criar um ‘Data Frame’ só com os ativos de renda variável. Isso está traduzido nas linhas do Código 5.16, criando `rendvar` com `.concat` com argumento `join = ‘inner’`. Na situação de se ter escolhido `join = ‘outer’` desde a criação do ‘Data Frame’ de criptomoedas, e aqui desejar-se manter o mesmo número de linhas, dever-se-ia usar também as linhas 61 e 62.

Código 5.16: Concatenação dos ativos da Bovespa e criptomoedas

```
60 rendvar = pd.concat([bolsa, criptos], axis = 1, join = 'inner')
61 rendvar.fillna(method = 'ffill', inplace = True)
62 rendvar.fillna(method = 'bfill', inplace = True)
```

5.2.4 Visualização do histórico da carteira

Diante de tantos dados armazenados vem a necessidade de visualização gráfica para facilitação de interpretação. Uma primeira apresentação pode ser a de comportamento dos preços dos ativo no decorrer do tempo adotado. Para ver isso em todos os ativos da carteira devemos primeiro criar um ‘Data Frame’ com todos os ativos dela. De modo análogo o que já fizemos antes, concatenamos os dados `fixa` aos de `rendvar`, com o `.merge`, como se apresenta no código 5.17. Os dados `cart` neste caso terão o mesmo número de linhas que os do tesouro direto, pois estes têm um número menor de datas de registro; isso porque se usou o argumento `how = ‘inner’`. As linhas 67 e 68 justificam-se uso apenas na mesclagem com `how = ‘outer’`.

Código 5.17: Junção de todo histórico dos ativos da carteira

```
63 cart=rendvar.merge(tessel2026[['Date', 'Tes. Selic 2026']], on='Date', how='inner')
64 cart=cart.merge(tespre2026[['Date', 'Tes. Pre. 2026']], on='Date', how='inner')
65 cart=cart.merge(tesipc2029[['Date', 'Tes. IPCA+ 2029']], on='Date', how='inner')
66 cart=cart.set_index('Date')
67 cart.fillna(method = 'ffill', inplace = True)
68 cart.fillna(method = 'bfill', inplace = True)
69 print(cart.head().to_string())
```

A Figura 5.9a da página 86 mostra o resultado da linha 69 dos dados `cart`. E com o conjunto de comandos dado no código 5.18 a seguir, se obteve a Figura 5.9b.

Código 5.18: Comandos para plotagem de gráficos de histórico de preços dos ativos

```
70 # importação do plotly para plotagem no navegador
71 import plotly.express as px
72 import plotly.graph_objects as go
73 fig = go.Figure()
74 fig = px.line(cart)
75 fig.update_layout(
76     title="Histórico dos preços dos ativos da carteira",
77     xaxis_title="2023",
78     yaxis_title="Preço (R$)",
79     legend_title="Ativo",
80     font=dict(
81         family="Courier New, monospace",
82         size=18,
83         color="Black"
84     )
85 )
86 fig.show()
```

Tal gráfico 5.9b comporta-se confuso por conta de apresentar os valores de ‘BTC’ muito maiores que dos demais, forçando a extensão vertical do gráfico compreender um intervalo muito grande, com os demais ativos “embaralhados” numa pequena extensão. O que se costuma fazer para mensurar melhor o comportamento é normalizá-los. Existem vários tipos de normalizações, e uma delas é dividir o valor em qualquer data pelo valor em uma data em específico. Fizemos a normalização pelo valor do primeiro dia, cujo Código 5.19 representa tal operação.

Código 5.19: Normalização do histórico dos ativos pelo valor na primeira data

```
87 cartnorm = cart / cart.iloc[0]
```

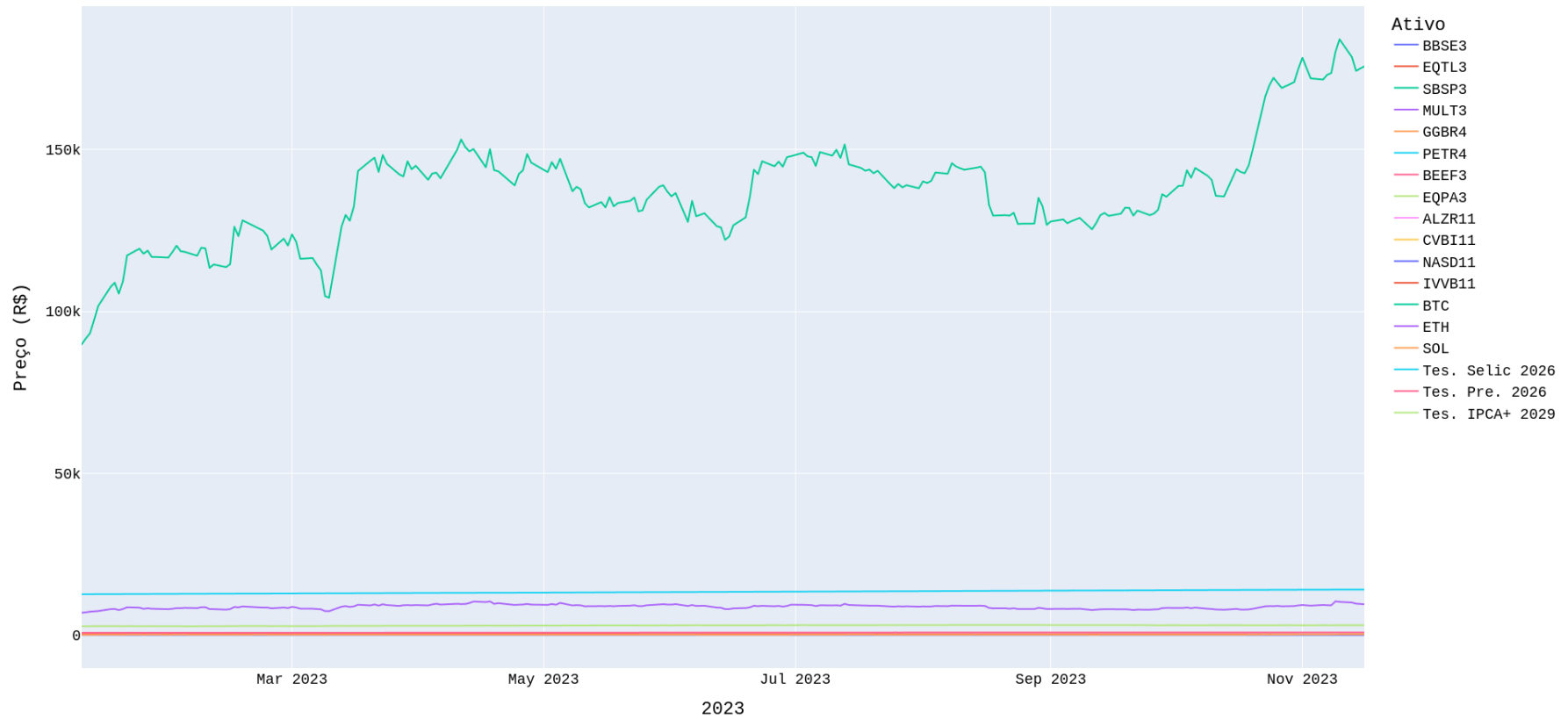
Figura 5.9: 'Data Frame' final e histórico de cotações da carteira no período de 09/01/2023 a 16/11/2023.

(a) 'Data Frame' do histórico dos ativos

Date	BBSE3	EQTL3	SBSP3	MULT3	GGBR4	PETRA	BEEF3	EQPA3	ALZR11	CVBI11	NASD11	IVVB11	BTC	ETH	SOL	Tes. Selic 2026	Tes. Pre. 2026	Tes. IPCA+ 2029
2023-01-09	30.380411	25.617401	53.507816	20.554981	28.164137	19.206923	12.589601	6.665839	106.402611	81.935326	8.15	225.449997	89841.681057	6904.248098	85.170591	12619.38	692.35	2705.45
2023-01-10	30.840313	25.844454	53.663391	21.184610	28.946728	19.383945	12.683904	6.775442	106.974968	82.436363	8.13	225.500000	91657.585523	7022.021886	85.092725	12625.50	695.98	2720.45
2023-01-11	31.020668	27.009331	54.285690	22.007973	29.237923	19.536829	13.664666	6.775442	106.890518	80.104210	8.20	226.199997	93230.970329	7214.890599	85.041145	12631.76	699.99	2732.99
2023-01-12	30.921474	27.196894	53.643944	21.620510	29.347120	19.818455	13.740109	6.775442	107.068794	80.322845	8.16	224.250000	97338.772805	7314.435907	85.729338	12637.99	704.96	2763.58
2023-01-13	30.840313	26.732918	53.186947	21.223358	29.510920	19.770176	14.381379	6.675802	107.791290	80.167976	8.20	224.850006	101692.129670	7414.413075	93.411886	12644.18	705.85	2754.94

(b) Gráfico do histórico dos ativos

Histórico dos preços dos ativos da carteira



Fonte: do autor

A Figura 5.10a apresenta a situação com normalização no valor na primeira data do histórico de cada ativo (`cartnorm`). Apesar das cores repetirem-se para ativos diferentes, o usuário pode passar com o cursor em qualquer ponto de uma curva e verificar qual ativo se refere ela e o valor normalizado naquela data⁹.

Na figura mostramos que **PETRA4** alcançou em 18 de outubro de 2023, duas vezes o valor do primeiro dia do histórico. Tal ativo foi o que “performou” melhor na carteira do Bovespa, tendo até 16/11/2023 com 1,85 do valor inicial. O tesouro **Selic**, até essa data, com 1,11, o tesouro **Prefixado** com 1,17, e o **IPCA+** com 1.13 do valor no início. Enquanto os piores, proporcionalmente ao valor da primeira data de registro, foram **GGBR** ao valor de 0,86, e **BEEF3** com 0,57.

O recurso de interativamente avaliar valores em gráfico, faz do *plotly* uma ferramenta prática. Por exemplo, na Figura 5.10b podemos destacar que a criptomoeda **BTC** apresentou uma performance melhor que **PETRA4**, mesmo que a primeira e a última tenham retornado porcentagens semelhantes do valor de início. Podemos notar que a volatilidade em **PETRA4** foi maior que de **BTC**, tendo por vezes valores negativos dentro do período.

A volatilidade é característica de muitos ativos de renda variável, principalmente os criptoativos, comum de serem classificados como de alto risco! Um exemplo disso está no desempenho de **SOL**, que apesar de se mostrar em 16/11/2023 ao valor de 3,3 do valor do dia 09/11/2023, oscilou até com valores negativos relativos a essa primeira data.

Também é possível observar o comportamento mediante os valores investidos em cada ativo. Para isso construímos um novo ‘Data Frame’, `histcart`, cujos valores de investimento de cada ativo é multiplicado pelo valor que cada um tem no `cartnorm`. A sintaxe envolvendo essa operação é a do Código 5.20 a seguir:

Código 5.20: Dados com valores da carteira no decorrer do tempo

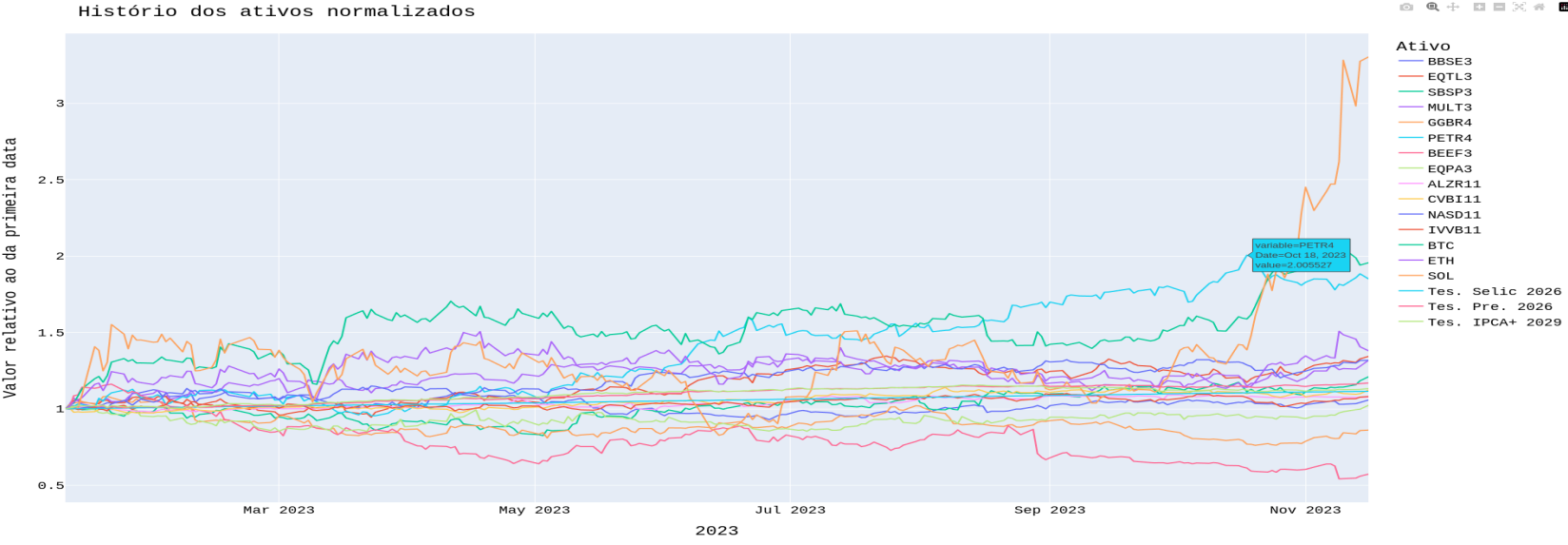
```
88 histcart = pd.DataFrame()
89 for ativo in df['Ativos']:
90     histcart[ativo] = cartnorm[ativo] * df.loc[df['Ativos'] == ativo]['Valor
investido (em reais)'].mean()
```

Do Código 5.20, cada ativo da carteira `df` é tomado pelo *loop for*, que em cada iteração realiza a operação do valor investido sendo multiplicado pelo histórico normalizado daquele ativo. O procedimento de localização foi aplicado semelhante ao que fizemos anteriormente com o `.loc`, mas aqui acrescentamos `.mean()` porque `.loc` retorna um ‘Data Frame’ e não um valor numérico¹⁰. A Figura 5.11 representa o histórico com `histcart`.

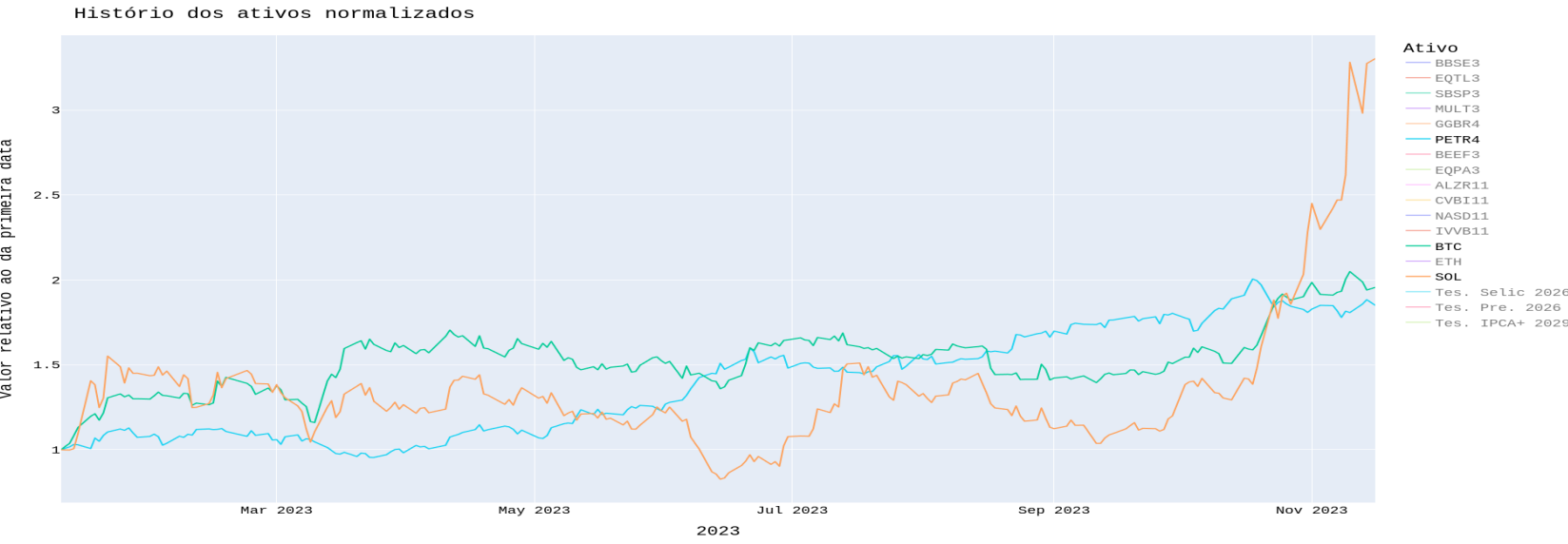
⁹Ao clicar num ativo da legenda que fica ao lado do gráfico, a respectiva curva do ativo desaparece. Isso pode facilitar visualizações dos demais ativos!

¹⁰Qualquer comando que retorna um valor numérico poderia ser usado, por exemplo, `.min()` ou `.max()`

Figura 5.10: Gráficos de históricos dos ativos.
(a) Ativos normalizados pelo valor na primeira data

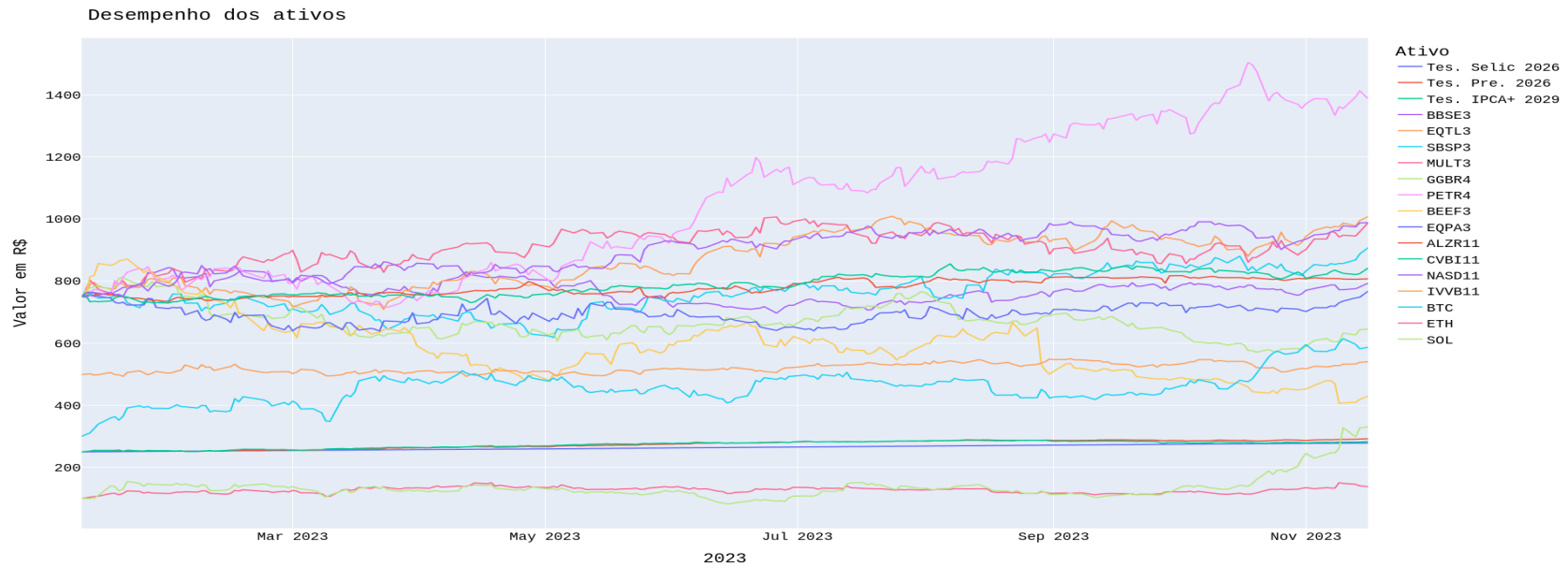


(b) Curvas dos ativos **PETR4**, **BTC** e **SOL**



Fonte: do autor

Figura 5.11: Histórico de desempenho de valores dos ativos



O histórico de cada ativo e o total da carteira é calculado como se apresenta no Código 5.21 abaixo. Primeiramente criamos o ‘Data Frame’ (inicialmente de uma só coluna) `histcartT` sendo a soma dos valores de cada linha de `histcart`¹¹. Esses dados são concatenados com os de histórico da carteira pela linha 92, e em seguida calcula-se os retornos simples de cada coluna de `histcartT`, ou seja, de cada ativo e do total armazenado na última coluna.

Código 5.21: Histórico dos rendimentos diários de cada ativo, e o retorno da carteira

```

91 histcartT = pd.DataFrame({'Total': histcart.sum(axis=1)})
92 histcartT = pd.concat([histcart, histcartT], axis = 1, join = 'inner')
93 retcartT = histcartT.iloc[-1] / histcartT.iloc[0] - 1

```

¹¹. `sum(axis=0)` faria a soma de valores de cada coluna.

De modo análogo podemos criar e obter os históricos de cada categoria da carteira. Os códigos 5.22 e 5.23 são justamente para criação dos ‘Data Frames’ dos históricos de somente renda fixa e variável da carteira, respectivamente; tendo a última coluna as somas dos rendimentos diários. Nas últimas linhas se tem as determinações dos retornos ponderados de cada categoria de ativos.

Código 5.22: Histórico dos rendimentos diários, e o retorno dos ativos de renda fixa

```

94 histfixa = histcart.loc[:, histcart.columns[:3]]
95 histfixT = pd.DataFrame({'Total': histfixa.sum(axis=1)})
96 histfixT = pd.concat([histfixa, histfixT], axis = 1, join = 'inner')
97 retfixT = histfixT.iloc[-1] / histfixT.iloc[0] - 1

```

Código 5.23: Histórico dos rendimentos diários, e o retorno dos ativos de renda variável

```

98 histvari = histcart.loc[:, histcart.columns[3:]]
99 histvarT = pd.DataFrame({'Total': histvari.sum(axis = 1)})
100 histvarT = pd.concat([histvari, histvarT], axis = 1, join = 'inner')
101 retvarT = histvarT.iloc[-1] / histvarT.iloc[0] - 1

```

Resultante da execução de `print(retcartT)`, `print(retfixT)` e `print(retvarT)`, separadamente, obtemos os retornos, que estão registrados na Figura 5.12.

Esses resultados estão na Figura 5.12 a seguir, em que temos com 20,14% o retorno de investimento na nossa carteira.

Figura 5.12: Retornos de toda carteira e de suas categorias.

(a) Da carteira		(b) Da renda fixa		(c) Da renda variável	
Tes. Selic 2026	0.114288	Tes. Selic 2026	0.114288	BBSE3	0.057918
Tes. Pre. 2026	0.169105	Tes. Pre. 2026	0.169105	EQTL3	0.343228
Tes. IPCA+ 2029	0.133043	Tes. IPCA+ 2029	0.133043	SBSP3	0.209730
BBSE3	0.057919	Total	0.138812	MULT3	0.317442
EQTL3	0.343228	dtype: float64		GGBR4	-0.139331
SBSP3	0.209730			PETR4	0.850895
MULT3	0.317442			BEEF3	-0.427305
GGBR4	-0.139331			EQPA3	0.023127
PETR4	0.850895			ALZR11	0.076571
BEEF3	-0.427305			CVBI11	0.122715
EQPA3	0.023127			NASD11	0.317791
ALZR11	0.076571			IVVB11	0.080727
CVBI11	0.122715			BTC	0.956160
NASD11	0.317791			ETH	0.380548
IVVB11	0.080727			SOL	2.302867
BTC	0.956160			Total	0.206502
ETH	0.380548				
SOL	2.302867				
Total	0.201425				

Fonte: do autor.

Aproveitamos esses resultados para registrar os retornos da carteira, da renda fixa e da renda variável considerando-se seus pesos, pelas variáveis `ret_car`, `ret_fix` e `ret_var`, que são assinaladas no Código 5.24. Seus respectivos valores foram 20,14%, 13,88% e 20,65%. Lembremo-nos que `ret_fix` será levado em conta no cálculo de *risk free* mais adiante.

Código 5.24: Retornos da carteira, da renda fixa e da renda variável, em todo o período.

```
102 ret_car = retcartT [ 'Total' ]
103 ret_fix = retfixT [ 'Total' ]
104 ret_var = retvarT [ 'Total' ]
```

5.3 Retornos dos ativos de renda variável

Vimos no capítulo 3 que a construção de uma carteira ótima de investimentos é calculada com retornos, simples ou logarítmico, sobre os ativos de renda variável. Eles podem ser diários, semanais, mensais, anuais, etc. Aqui vamos explorar os diários. Então, inicialmente criamos um ‘Data Frame’ para esse medida, a qual está codificada no Código 5.25 abaixo. Na linha 105, o ‘Data Frame’ `rendvar` foi dividido por ele mesmo mas no valor da data do dia anterior, depois subtraído por 1. O comando `.shift` dá o acesso ao valor com localização do argumento que está nele, no caso nosso, de um dia anterior apenas. Decorrente no entanto do cálculo do retorno diário, o primeiro valor (data início), fica com ‘NaN’ e consequentemente, na linha 106 fazemos o ajuste para se adotar valor nulo.

Código 5.25: Cálculo do retorno diário de cada ativo

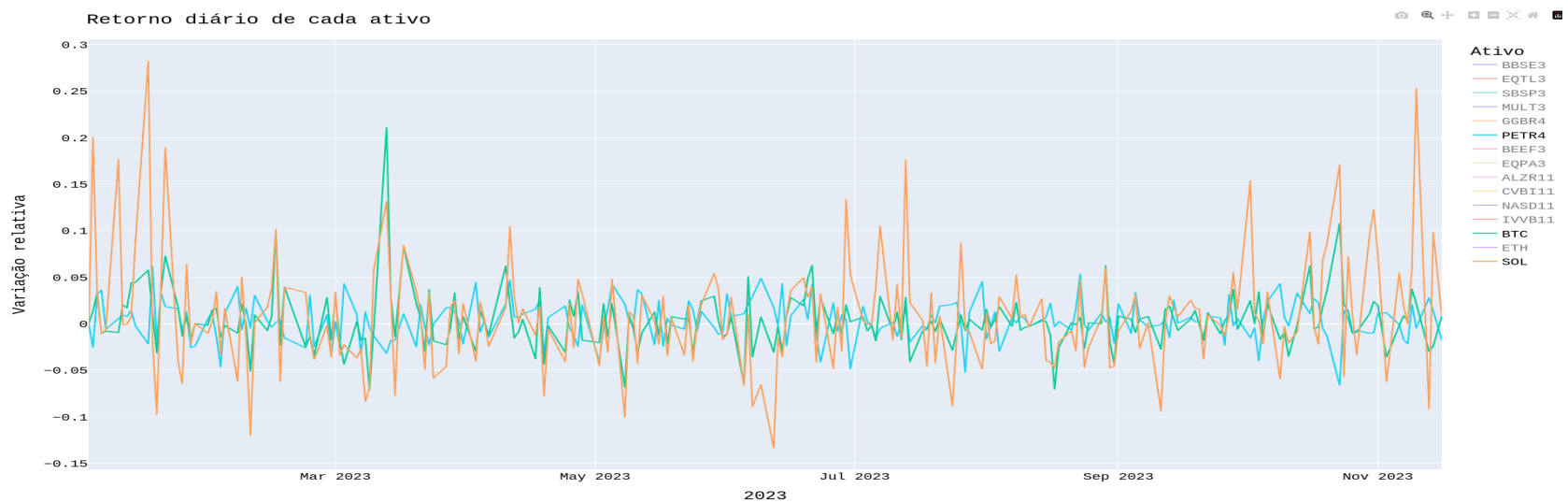
```
105 retdiario = rendvar/rendvar.shift(1) - 1
106 retdiario.fillna(value = 0, inplace = True)
```

Com comandos análogos aos do Código 5.18¹² podemos plotar em gráfico de linha esses retornos diários dos ativos (`retdiario`). As Figuras de 5.13 podem fomentar algumas discussões. Uma é a alta volatilidade que se pode observar em alguns ativos, como da bolsa e criptomoedas. Nessa Figura 5.13a foram deixadas somente **PETR4**, **BTC** e **SOL**, e nos dias 16, 17 e 18 de janeiro de 2023 ocorreram variações relativas de retorno de 0,28, -0,09 e 0,18, consecutivas em **SOL**!

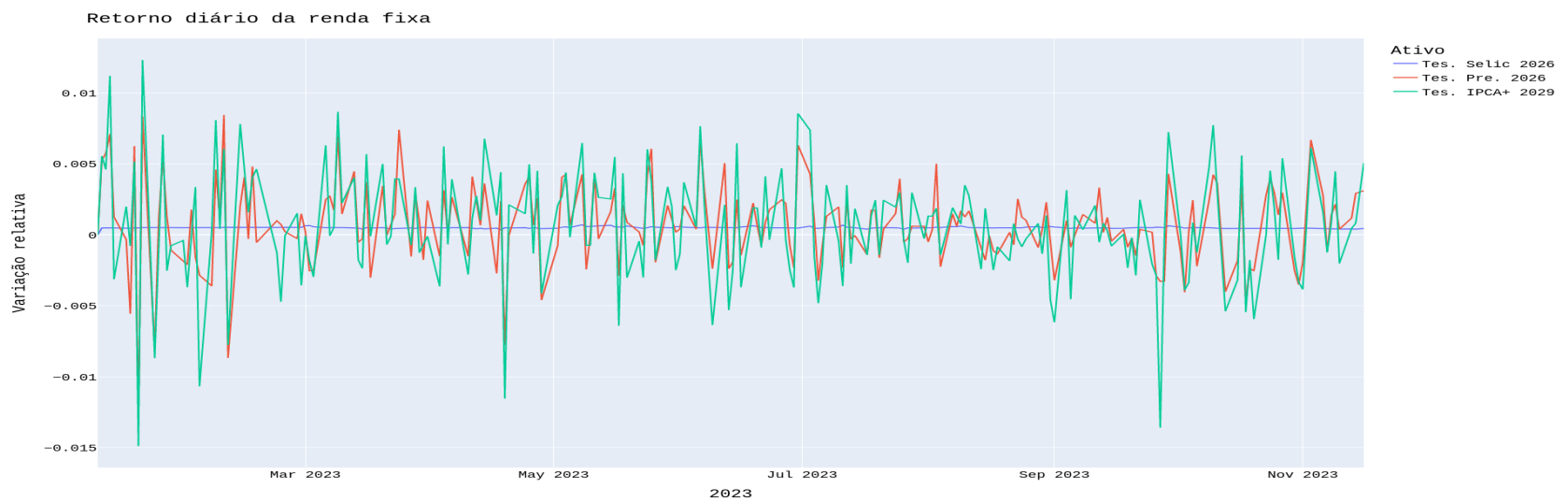
¹²Só precisamos adequar as linhas 74, 76 e 78. E não mais precisamos chamar 71 e 72.

Figura 5.13: Retornos diários de alguns ativos da carteira.

(a) Retorno de **PETR4**, **BTC** e **SOL**



(b) Retorno diários dos ativos do Tesouro



Fonte: do autor

Outro fato interessante foi a “concordância” de variação dos retornos diários entre ‘BTC’ e ‘SOL’, e a “discordância”, em muitos casos, de variações destes com ‘PETR4’. Isso se mostraria nas medidas de correlação? Comparativamente, em 5.13b percebemos a baixa volatilidade na renda fixa, sendo a maior no ativo ‘IPCA+’ que é atrelado ao ‘DI’, variável bem mais volátil que a poupança.

Elementos de estatística básica, como média, variância, desvio-padrão e correlação tem funções intrínsecas no *Pandas*. Os simples comandos para isso estão nas linhas de 5.26:

Código 5.26: Determinação da média, variância, desvio-padrão e correlação dos retornos diários

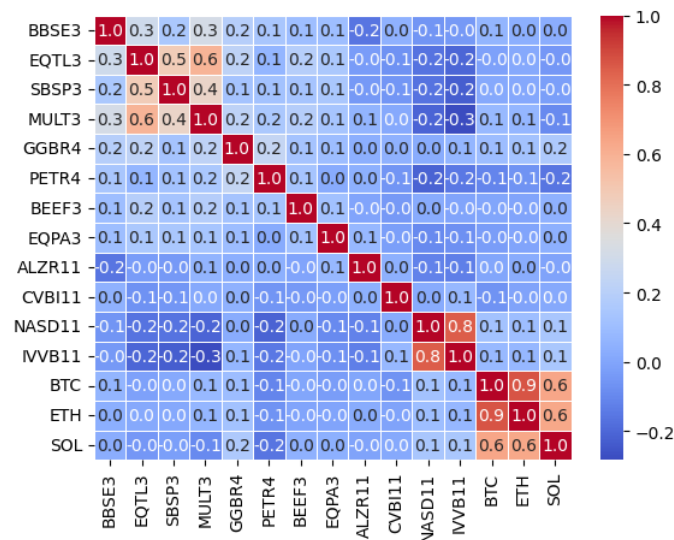
```
107 retdiariomed = retdiario.mean()
108 covar = retdiario.cov()
109 desvio = retdiario.std()
110 corre = retdiario.corr()
```

A correlação nos permite, por exemplo, verificar as “concordâncias” ou não existentes nos retornos entre ativos. Uma maneira de visualizar a matriz de correlação é usando um mapa de “calor” dos seus valores. Aqui abaixo obtemos o gráfico 5.14, com as entradas dos elementos que relacionam os ativos o valor da correlação existente entre eles. Neles vemos que a diagonal é justamente a variância de um ativo com ele mesmo.

Código 5.27: Matriz correlação por mapa de calor

```
111 plt.figure()
112 sea.heatmap(corre, annot=True, fmt='.1f', linewidth=.5)
```

Figura 5.14: Matriz correlação por mapa de calor nos ativos de renda variável



Fonte: do autor

Uma pequena investigação sobre a figura interativa, ilustrada em 5.13a, pode por exemplo verificar que para o valor de coeficiente de correlação alto dado entre **BTC** e **ETH**, 0,9; reflete no comportamento de variação dos retornos muito semelhantes e praticamente de mesmas amplitudes. Sobre os ativos **BTC** e **SOL**, o coeficiente é positivo de 0,6; poderemos notar que as variações acontecem no mesmo sentido na maioria das vezes, mas as amplitudes delas são bem distintas. Em sentido oposto podemos verificar que **MUTL3** e **IVVB11**, com coeficiente -0,3, têm variações em sentido contrário uma da outra em maioria dos retornos diários; com as suas amplitudes não tão iguais em comprimento. Portanto, o coeficiente de correlação entre ativos pode sugerir decisões de investimento.

5.4 Geração de portfólios aleatórios de uma carteira

Como foi dito no capítulo 3 o modelo de Markowitz é analisado sobre o comportamento de portfólios tendo como variáveis o Risco (desvio-padrão) e Retorno Esperado (de retornos simples, ou logarítmico, ponderado pelos pesos dos ativos). A carteira ótima é aquela com informações de risco e retorno no ponto de tangência da reta, de coeficiente linear o *risk free* e angular o máximo índice Sharpe, com a fronteira eficiente de Markowitz.

A determinação da fronteira de Markowitz já se constitui por si só um problema de otimização, e que pode ser atacado por diversos métodos dentro dos ramos de otimização não linear computacional. Tendo em vista os conteúdos que eles levam não pertencer aos de ensino básico, restringimos as buscas de ‘carteira ótima’ como aproximada pelo ponto de encontro da reta que passa no *risk free* e o ponto de máximo índice Sharpe que se obtém com pontos gerados aleatoriamente.

Esses pontos gerados aleatoriamente são sobre os retornos esperados, e por isso têm restrições. Primeiro que os retornos, simples ou logarítmico, adotados devem assumir valores condizentes com o histórico de retorno. O comum é então adotar o retorno médio de cada ativo¹³! Em segundo lugar, os pesos, que são a parte gerada aleatoriamente, devem ser tais que sua soma seja a unidade.

Podemos então criar portfólios aleatórios, registrando em cada um, o risco (ou volatilidade) e o retorno esperado, com as fórmulas 3.6 e 3.4, respectivamente. Tal registro pode ser feito em *arrays* do *numpy* para posteriormente serem plotados para visualização dos portfólios aleatórios. Como a determinação de nossa “carteira ótima” será com o máximo índice Sharpe encontrado sobre os pontos gerados aleatoriamente desses portfólios, aproveitaremos nessa ocasião para armazenar esse índice em um *array*. O Código 5.28 abaixo transcreve as

¹³É também possível levar em conta os retornos médios adicionados a, valores aleatórios em distribuição normal que são multiplicados pelos desvios-padrões desses ativos.

ações que acabamos de discutir.

Código 5.28: Programa de geração de portfólios aleatórios nos ativos de renda variável.

```

113 nat = len(retdiariomed)
114 nrt = len(retdiario.index)
115 npo = 100000
116 ret = []
117 vol = []
118 sha = []
119 rf = (1+ret_fix)**(1/nrt) - 1
120 pes = np.zeros((npo, nat))
121 for i in range(npo):
122     w = np.random.random(nat)
123     w /= np.sum(w)
124     pes[i, :] = w
125     p = np.asmatrix(retdiariomed)
126     w = np.asmatrix(w)
127     C = np.asmatrix(covar)
128     vol.append(np.sqrt(w * C * w.T))
129     ret.append(w * p.T)
130     sha.append((ret[-1] - rf) / vol[-1])

```

Na linha 113 criamos uma variável com o número de ativos da renda variável, enquanto `nrt` é o número de retornos contidos nesse ‘Data Frame’. A linha 115 assume a variável `npo` como sendo o número de portfólios aleatórios que serão construídos. Já que estaremos determinando a carteira “ótima” pelos pontos dos portfólios aleatórios, é crucial que se crie uma quantidade muito grande deles.

As linhas 116, 117 e 118 nada mais são que as criações dos *listas*, ainda vazios, para o retorno esperado, volatilidade (risco) e índice Sharpe. Como este depende do *risk free*, ele é calculado na linha 119. A fim de posteriormente retornar informações dos pesos que um determinado portfólio assume, criamos uma matriz do *numpy*, inserida com zeros na linha 120.

Na linha 121 iniciamos o *loop* para criação dos portfólios, cujos pesos foram criados, na quantidade de ativos com números aleatórios entre 0 e 1, com o comando `np.random.random(nat)` (linha 122). Para garantir que eles tenha soma 1, foram então todos divididos pela sua soma na linha 123; enquanto tais valores foram armazenados para posterior uso em 124.

Nas linhas 125, 126 e 127 houveram as conversões das variáveis `retdiariomed`, `w` e `covar` para valores numéricos em *arrays*. Recordemos que `retdiariomed` e `covar` eram ‘Data Frames’, enquanto `w` antes, era da forma `(nat,)`, não sendo um *array* de uma coluna ou uma linha

Nas linhas 128 e 129 se tem os cálculos do risco e o retorno esperado, atentando-se às dimensões que possibilitem os produtos. Já na linha 130 o índice Sharpe é tido sobre os últimos valores do retorno esperado e volatilidade, e o *risk free*.

Decorrente de todos os portfólios gerados podemos extrair, por exemplo, o risco mínimo e retorno esperado e pesos associados; ou, os índices Sharpe de cada um. Apesar de riscos, retornos esperados e índices Sharpe estarem em *listas*, decidimos recuperar elementos de interesse não por *listas*, mas no ‘Data Frame’ deles. A sintaxe disso, semelhante ao que já fizemos antes, está na linha 131 do Código 5.29 a seguir.

Código 5.29: ‘Data Frame’ dos riscos, retornos esperados e índices Sharpe de portfólios aleatórios.

```

131 port = pd.DataFrame({'Retorno':ret, 'Risco (Volatilidade)':vol, 'Sharpe': sha})
132 riscomin = port['Risco (Volatilidade)'].min().item()
133 iminvol = port.index[port['Risco (Volatilidade)'] == riscomin].item()
134 retormin = port['Retorno'][iminvol].item()
135 pesosmin = pes[iminvol, :]

```

Nesse código também se determinou o risco mínimo dos portfólios aleatórios, e seu índice no ‘Data Frame’, para então referenciá-lo para registro do retorno esperado e pesos associados. Uma sintaxe não usada antes foi o comando `.item()`. Sua razão de uso é porque o comando `.min()` em `port` retorna uma matriz, ainda de um valor apenas. O `.item()` tranforma-o em valor numérico.

Resultante dessas determinações podemos então anunciá-las para realizar possíveis comparações. Mas antes disso nos pareceu conveniente armazenar em *lista* os *tickers* e *tokens* dos ativos de renda variável para apresentarmos alocações mediante pesos encontrados nos resultados. Essa construção se apresenta nas seguintes linhas de código:

Código 5.30: Ativos da Renda Variável armazenados em *lista*

```

136 ativvar = []
137 for i in range(nat):
138     ativvar.append(rendvar.columns[i])

```

Agora com o Código 5.31 abaixo podemos visualizar os resultados, e cuja apresentação está na Figura 5.15. Nesse caso usamos a semente de geração de números aleatório `np.random.seed(212)` imediatamente antes da linha 121 do Código 5.28 com intuito de disponibilização de verificação de um leitor interessado.

Código 5.31: Apresentação de risco mínimo, com retorno esperado e pesos associados, em portfólios aleatórios.

```

139 print('Risco mínimo retornado pelos portfólios aleatórios:')
140 print(riscomin)
141 print('Retorno mínimo no período, obtido pelos portfólios aleatórios:')
142 print(retormin*nrt)
143 print('Pesos associados ao risco mínimo obtido pelos portfólios aleatórios:')
144 print(pesosmin)
145 print('Carteira renda variável com o mínimo risco dos portfólios aleatórios:')
146 for peso, ativo in zip(pesosmin,ativvar):
147     print(f'{round(peso*100,3)}% de renda variável deve ser alocada a {ativo}.')
```

Observemos que na linha 142 fizemos a multiplicação do retorno esperado diário pelo número de retornos diários que foram armazenados no ‘Data Frame’ `retdiario`, o qual teve valor de 19,17%. Notemos que esse valor é próximo daquele retorno de renda variável `ret_var`, que obtivemos pela nossa carteira! No entanto isso pode ser apenas coincidência!

O *loop* `for` da linha 146 é tomando varrendo-se a *tupla* `pesosmin` e `ativvar`, construída pelo comando `zip`. Isso significa que o `for` tomará ‘contadores’ `peso` e `ativo` variando-se nos elementos dessa *tupla* `(pesosmin,ativvar)`.

Figura 5.15: Risco mínimo resultante dos portfólios aleatórios, retorno esperado e pesos associados, com também a alocação da carteira de renda variável sob tal risco.

```

Risco mínimo retornado pelos portfólios aleatórios:
0.004865610184334127
Retorno mínimo no período, obtido pelos portfólios aleatórios:
0.19169818363747737
Pesos associados ao risco mínimo retornado pelos portfólios aleatórios:
[0.14857759 0.06196165 0.05029768 0.09061358 0.0192745 0.03648005
 0.01670027 0.01040724 0.13202053 0.1496567 0.1142537 0.14891551
 0.0063391 0.00753846 0.00696342]
Carteira de renda variável com o mínimo risco dos portfólios aleatórios:
-----
14.858% de renda variável deve ser alocada a BBSE3.
6.196% de renda variável deve ser alocada a EQTL3.
5.03% de renda variável deve ser alocada a SBSP3.
9.061% de renda variável deve ser alocada a MULT3.
1.927% de renda variável deve ser alocada a GGBR4.
3.648% de renda variável deve ser alocada a PETR4.
1.67% de renda variável deve ser alocada a BEEF3.
1.041% de renda variável deve ser alocada a EQPA3.
13.202% de renda variável deve ser alocada a ALZR11.
14.966% de renda variável deve ser alocada a CVBI11.
11.425% de renda variável deve ser alocada a NASD11.
14.892% de renda variável deve ser alocada a IVVB11.
0.634% de renda variável deve ser alocada a BTC.
0.754% de renda variável deve ser alocada a ETH.
0.696% de renda variável deve ser alocada a SOL.
-----
```

Nosso objetivo principal passa ser agora a determinação da carteira com risco \times retorno esperado “ótimo”. Vamos tomá-la com aproximação do máximo índice Sharpe sobre os portfólios gerados aleatoriamente. Então, aqui podemos encontrá-lo por:

Código 5.32: Máximo índice Sharpe dos portfólios gerados aleatoriamente.

```
148 maxsha = port[ 'Sharpe' ].max().item()
149 imaxsha = port.index[port[ 'Sharpe' ] == maxsha].item()
```

Tendo então o índice de `port` em que se encontra o referido índice Sharpe (`imaxsha`), podemos facilmente apresentar o risco e retorno esperado ótimo pelos comandos do Código 5.33:

Código 5.33: Risco \times retorno esperado ótimo determinado pelos portfólios aleatórios.

```
150 riscooti = port[ 'Risco (Volatilidade)' ][imaxsha].item()
151 retoroti = port[ 'Retorno' ][imaxsha].item()
152 pesosport = pes[imaxsha, :]
153 print( 'Maximo Indice Sharpe, retornado pelos portfolios aleatorios:')
154 print( port[ 'Sharpe' ][imaxsha].item())
155 print( 'Risco ótimo retornado pelos portfólios aleatórios:')
156 print( port[ 'Risco (Volatilidade)' ][imaxsha].item())
157 print( 'Retorno otimo no período, obtido pelos portfolios aleatorios:')
158 print( port[ 'Retorno' ][imaxsha].item()*nrt)
159 print( 'Pesos advindos do máximo Sharpe, obtido pelos portfólios aleatórios:')
160 print( pesosport)
161 print( 'Carteira renda variável do máximo Sharpe dos portfólios aleatórios:')
162 print( '_____')
163 for peso, ativo in zip( pesosoti, ativvar ):
164     print( f' {round(peso*100,3)}% de renda variável deve ser alocada a {ativo}.' )
165 print( '_____')
```

Os resultados da execução desse conjunto de comandos estão na Figura 5.16 a seguir. Observemos que nesse caso o retorno esperado no período de observações foi de 46,29%, sob o risco de 0,00829, usando a semente `np.random.seed(212)`. Esses números ficam com uma melhor interpretação quando os portfólios aleatórios, e os pontos de risco mínimo e o ‘otimizado’ são apresentados em gráfico.

Os comandos para plotagem estão no Código 5.34. Repare que primeiro foram transformadas para *arrays numpy* as *listas* `ret` e `vol`. Em seguida vem o comando `figure` que neste caso tomamos com argumento `figsize` para definir a largura e altura da figura. Os comandos `scatter` foram usados para inserir pontos definidos por suas coordenadas, configurando-se os marcadores (`marker`), seus tamanhos (`s`), e cores (`color`). Na última inserção de pontos, fizemos o requerimento de que pudessem ser retratados pelo mapa de cores (`cmap`), variando-se

em função do valor do índice Sharpe ($c = (\text{ret} - \text{rf}) / \text{vol}$), com transparência definida por alpha.

Figura 5.16: Risco mínimo resultante dos portfólios aleatórios, retorno esperado e pesos associados, com também a alocação da carteira de renda variável sob tal risco.

```
Máximo Índice Sharpe, retornado pelos portfólios aleatórios:
0.18328718303292038
Risco ótimo retornado pelos portfólios aleatórios:
0.008292625015109602
Retorno ótimo, diário, pelos portfólios aleatórios:
0.46288912954864375
Pesos advindos do máximo Sharpe, retornado pelos portfólios aleatórios:
[0.0410905  0.05521896 0.02794044 0.13934275 0.00151134 0.15532327
 0.02322532 0.00227847 0.11226575 0.07017535 0.14819773 0.05759818
 0.08442714 0.00037276 0.08103204]
Carteira de renda variável do máximo Sharpe dos portfólios aleatórios:
-----
4.109% de renda variável deve ser dedicada a BBSE3.
5.522% de renda variável deve ser dedicada a EQTL3.
2.794% de renda variável deve ser dedicada a SBSP3.
13.934% de renda variável deve ser dedicada a MULT3.
0.151% de renda variável deve ser dedicada a GGBR4.
15.532% de renda variável deve ser dedicada a PETR4.
2.323% de renda variável deve ser dedicada a BEEF3.
0.228% de renda variável deve ser dedicada a EQPA3.
11.227% de renda variável deve ser dedicada a ALZR11.
7.018% de renda variável deve ser dedicada a CVBI11.
14.82% de renda variável deve ser dedicada a NASD11.
5.76% de renda variável deve ser dedicada a IVVB11.
8.443% de renda variável deve ser dedicada a BTC.
0.037% de renda variável deve ser dedicada a ETH.
8.103% de renda variável deve ser dedicada a SOL.
-----
```

Fonte: do autor.

Código 5.34: Comando para plotagem de portfólios aleatórios.

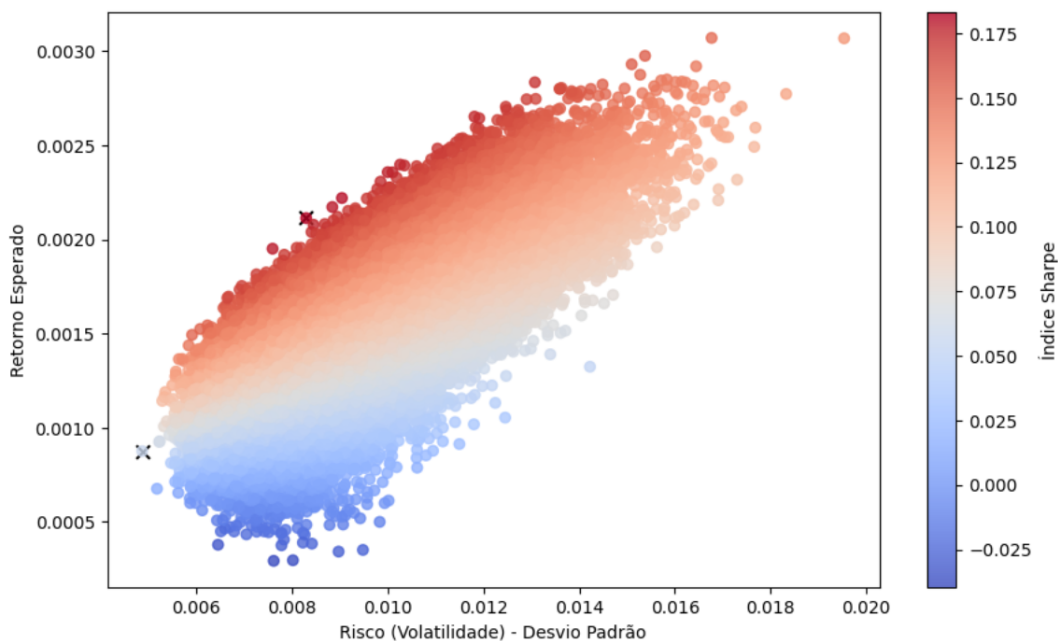
```
166 ret = np.array(ret)
167 vol = np.array(vol)
168 plt.figure(figsize = (10,6))
169 plt.scatter(riscooti , retoroti , marker = 'x' , s = 60 , color = 'k')
170 plt.scatter(riscomin , retormin , marker = 'x' , s = 60 , color = 'k')
171 plt.scatter(vol , ret , c = (ret - rf) / vol , marker = 'o' , cmap = 'coolwarm' ,
    alpha = 0.8)
172 plt.xlabel('Risco (Volatilidade) - Desvio Padrão')
173 plt.ylabel('Retorno Esperado')
174 plt.colorbar(label='Índice Sharpe')
175 plt.show()
```

Esses comandos geram a Figura 5.17a. No entanto, apesar do grande número de portfólios aleatórios gerados (cem mil), não podemos crer que os pontos que parecem delimitar a região gerada por todos compreenda à fronteira eficiente. Com a função `minimize` do pacote `optimize`, da biblioteca `scipy` fizemos a programação de busca do mínimo risco, do máximo

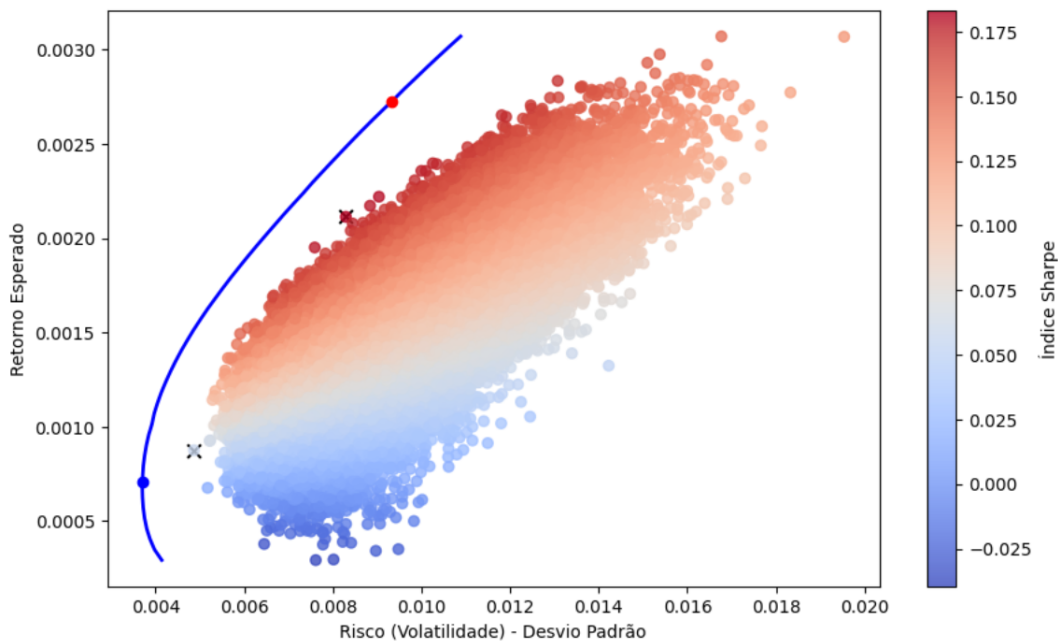
índice Sharpe, e da fronteira eficiente da carteira dos ativos de renda variável. Como resultado desses três itens de otimização encontramos os que se mostra na Figura 5.17b.

Figura 5.17: Portfólios aleatórios. Pontos de mínimo risco. Fronteira eficiente.

(a) Portfólios aleatórios, com risco mínimo e carteira 'ótima' demarcados



(b) Fronteira eficiente, com risco mínimo e carteira ótima obtidos por otimização



Fonte: do autor

Da otimização para o menor risco se obteve os valores apresentados na Figura 5.18a,

que são mensurações do que se percebe graficamente. Já para a otimização do máximo índice Sharpe se teve os resultados apresentados na Figura 5.18b

Figura 5.18: Resultados de mínimo risco e carteira ótima da renda variável obtidos por otimização.

(a) Risco mínimo obtido por otimização, retornos e carteira associados.

```

-----
Risco mínimo retornado por otimização:
0.003698424185167207
-----
Retorno mínimo, diário, retornado por otimização:
0.1546773198527225
-----
Pesos retornados pela otimização de menor risco:
[0.071 0.054 0.038 0.    0.    0.038 0.003 0.022 0.353 0.208 0.046 0.151
 0.015 0.    0.    ]
-----
Carteira de renda variável com o mínimo risco por otimização:
-----
7.073% de renda variável deve ser dedicada a BBSE3.
5.399% de renda variável deve ser dedicada a EQTL3.
3.813% de renda variável deve ser dedicada a SBSP3.
0.0% de renda variável deve ser dedicada a MULT3.
0.0% de renda variável deve ser dedicada a GGBR4.
3.836% de renda variável deve ser dedicada a PETR4.
0.334% de renda variável deve ser dedicada a BEEF3.
2.172% de renda variável deve ser dedicada a EQPA3.
35.3% de renda variável deve ser dedicada a ALZR11.
20.83% de renda variável deve ser dedicada a CVBI11.
4.649% de renda variável deve ser dedicada a NASD11.
15.101% de renda variável deve ser dedicada a IVVB11.
1.493% de renda variável deve ser dedicada a BTC.
0.0% de renda variável deve ser dedicada a ETH.
0.0% de renda variável deve ser dedicada a SOL.
-----

```

(b) Máximo índice Sharpe e carteira ótima.

```

-----
Máximo Índice Sharpe retornado por otimização:
0.22884925175146037
-----
Risco ótimo retornado pelo máximo Sharpe otimizado:
0.00932826644891466
-----
Retorno ótimo no período, pelo máximo Sharpe otimizado:
0.5975379765916521
-----
Pesos advindos do máximo Sharpe otimizado:
[0.00000000e+00 1.37990252e-01 1.24349868e-17 7.32037688e-02
 4.02749128e-17 3.23587527e-01 4.74967927e-17 0.00000000e+00
 1.49291317e-17 6.14755588e-02 2.86626959e-01 5.21134290e-17
 1.30405419e-02 2.22872299e-17 1.04075393e-01]
-----
Carteira de renda variável do máximo Sharpe por otimização:
-----
0.0% de renda variável deve ser dedicada a BBSE3.
13.799% de renda variável deve ser dedicada a EQTL3.
0.0% de renda variável deve ser dedicada a SBSP3.
7.32% de renda variável deve ser dedicada a MULT3.
0.0% de renda variável deve ser dedicada a GGBR4.
32.359% de renda variável deve ser dedicada a PETR4.
0.0% de renda variável deve ser dedicada a BEEF3.
0.0% de renda variável deve ser dedicada a EQPA3.
0.0% de renda variável deve ser dedicada a ALZR11.
6.148% de renda variável deve ser dedicada a CVBI11.
28.663% de renda variável deve ser dedicada a NASD11.
0.0% de renda variável deve ser dedicada a IVVB11.
1.304% de renda variável deve ser dedicada a BTC.
0.0% de renda variável deve ser dedicada a ETH.
10.408% de renda variável deve ser dedicada a SOL.
-----

```

Comparando os retornos esperados entre o que encontramos para o máximo índice Sharpe dos portfólios aleatórios e os encontrados por otimização temos o erro absoluto $|0,5975 - 0,4629| = 0,1346$, ou seja, de um pouco mais de 13%. Além disso, a composição dos ativos da carteira ótima de renda variável, obtida por otimização, tem apenas 7, dentre os 15 ativos de renda variável disponíveis com pesos significativos. Enquanto que aquela ‘ótima’ considerando-se os portfólios aleatórios tem pesos distribuídos por 12 dos ativos.

Mais quantitativamente, podemos comparar os rendimentos entre as carteiras considerando os pesos encontrados! Inicialmente foi reservado R\$ 9.250,00 para aplicação na renda variável. Então ao multiplicarmos esse valor para cada conjunto de pesos encontrados teremos os valores de aplicação de cada ativo, de cada carteira. Temos que os pesos encontrados pelos portfólios aleatórios foram armazenados em `pesoport`, enquanto que os encontrados por otimização, digamos, tenham sido sob a variável `pesosotim`. Então, recordando que o ‘Data Frame’ `cartnorm` tem o histórico de cada ativo, normalizado pelo valor na primeira data; para obtermos os rendimentos de cada carteira variável faríamos:

Código 5.35: Rendimentos das carteiras de renda variável

```
176 rendport = np.dot(cartnorm.iloc[-1][:15].values, pesosport * 9250)
177 rendotim = np.dot(cartnorm.iloc[-1][:15].values, pesosotim * 9250)
```

Do que resultam nos valores `rendport = 14.151,90` e `rendotim = 15.694,58` reais. A diferença relativa neste caso fica $|15.694,58 - 14.151,90|/15.694,58 = 0.0983$, ou seja, de 9,83%. Em termos absolutos a diferença de rendimento foi de R\$ 1542,62; valor pelo qual pode ser interpretado como bastante significativo, já que a aplicação inicial foi de R\$ 9250,00. Porém, aqui dedicamos um percentual (92,5%) em renda variável muito além do que se aconselha para investidores, mesmo para os mais arrojados/sofisticados!

Sobre as características afirmadas na subsubseção 3.3.5, podemos verificar que a carteira ótima não precisou ter muitos ativos mesmo. No entanto, apesar dos ativos retornados terem os coeficientes de correlação muito pequenos (pouco correlatos) entre eles; outros que têm essa mesma propriedade, possuíam praticamente peso nulo; portanto não se pode olhar aos coeficientes de correlação e simplesmente escolher ativos com tal característica!

5.5 Aproximação para uma carteira ótima.

Para concluir a construção da carteira, vamos determinar a *Capital Market Line* conforme apresentada na subsubseção 3.6.2. Lá vimos que é sobre ela que se tem um ponto responsável pelo peso w com a contribuição da aplicação em renda variável R_P , restando $1 - w$ para os ativos de renda fixa. O peso w apresentado na seção 3.6, é a razão, da diferença do retorno

esperado da carteira de renda variável ótimo e o *risk free*, com o produto do quadrado do risco ótimo pela constante informando a aversão ao risco do investidor. Para continuarmos fazendo comparação, teremos então:

Código 5.36: Determinação das carteiras ótimas por portfólios aleatórios e otimização

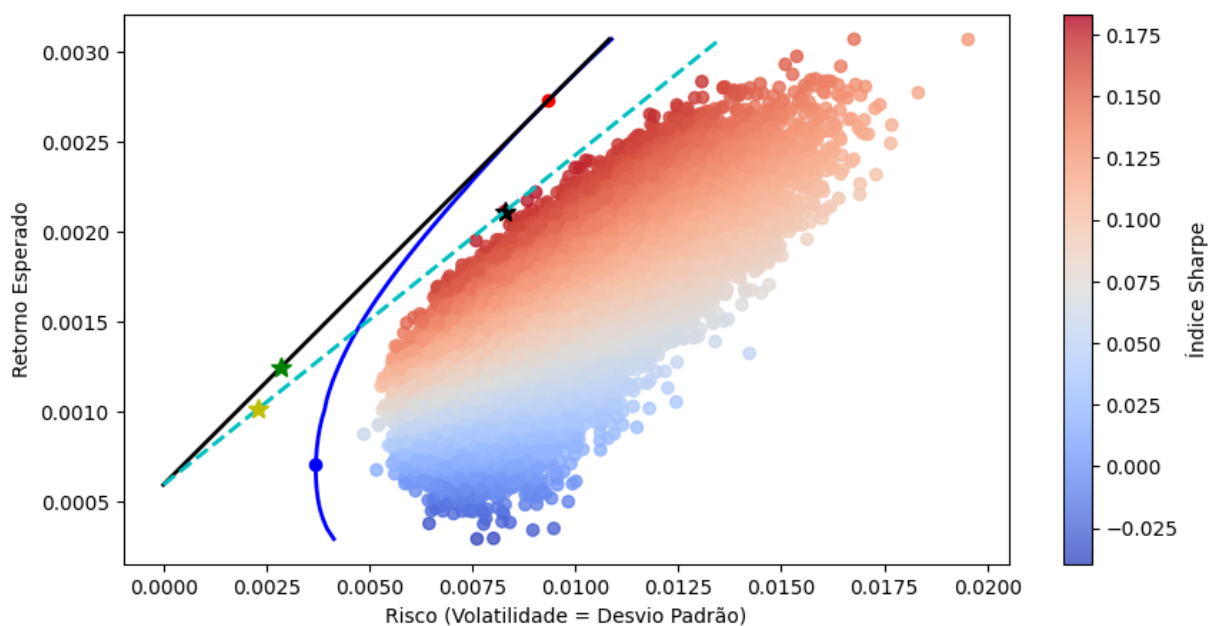
```

178 a = 80
179 wpa = (retorport - rf) / (a * riscoport**2)
180 wpo = (retorotim - rf) / (a * riscootim**2)
181 risa = wpa * riscoport
182 riso = wpo * riscootim
183 reta = wpa * retorport + (1 - wpa) * rf
184 reto = wpo * retorotim + (1 - wpo) * rf

```

Na linha 178 fazemos a atribuição um valor à constante de aversão ao risco do investidor. Nas linhas 179 e 180 fazemos os cálculos dos pesos para as carteira de renda variável no caso de risco e retorno ótimo obtidos pelos portfólios aleatórios **wpa**, e os obtidos por otimização **wpo**. Nos casos dos risco e retorno resultantes da otimização, eles estão na variáveis **riscootim** e **retorotim**. Associados aos pesos encontrados, calculamos então os riscos e retornos esperados dessas carteiras pelas linhas 181, 182, 183 e 184. Com os pares risco \times retorno esperado podemos então plotar os pontos referentes às carteiras ótimas, como estão mostrados na figura 5.19 abaixo, com o marcador estrela verde claro atribuído ao ponto dos portfólios aleatórios e o de estrela verde escuro da otimização.

Figura 5.19: Carteiras ótimas vindas dos portfólios aleatórios e de otimização.



Fonte: do autor

A interseção das CMLs acontece exatamente no valor `rf` de *risk free*, e traçadas daí até o pontos de máximo índice Sharpe encontrados em cada aproximação.

Com essas informações dos pesos podemos então avaliar que rendimentos elas teriam sobre o valor de R\$ 10.000,00 investidos, para um investidor com índice de aversão ao risco no valor de 80. Nas linhas do Código 5.37 fazemos os cálculos desses rendimentos, sendo levados em conta os pesos encontrados para os de renda variável (por portfólios e otimização), e os de renda fixa distribuídos igualmente entre os 3 ativos do tesouro.

Código 5.37: Rendimentos das carteiras ótimas pelos portfólios aleatórios e por otimização.

```

185 rvarport = np.dot(cartnorm.iloc[-1][:15].values, pesosport) * wpa * 10000
186 rfixport = np.dot(cartnorm.iloc[-1][15:].values, np.array([1/3, 1/3, 1/3])) * \
187         (1 - wpa) * 10000
188 rcarport = rvarport + rfixport
189 rvarotim = np.dot(cartnorm.iloc[-1][:15].values, pesosotim) * wpo * 10000
190 rfixotim = np.dot(cartnorm.iloc[-1][15:].values, np.array([1/3, 1/3, 1/3])) * \
191         (1 - wpo) * 10000
192 rcarotim = rvarotim + rfixotim

```

Os valores disto são `rvarport = 4226.91`, `rfixport = 8241.81` e `rcarport = 12468.71`, de rendimentos de renda variável, renda fixa e total com as ponderações obtidas dos portfólios aleatórios. Enquanto que são `rvarotim = 5203.16`, `rfixotim = 7895.82`, `rcarotim = 13098.98` foram resultantes da aplicação dos pesos obtidos por otimização.

Comparando os rendimentos das duas carteiras temos a diferença absoluta como sendo de $|13098.98 - 12468.71| = 630.26$ reais, diferença relativa percentual de $|13098.98 - 12468.71|/13098.98 \cdot 100$ correspondendo a somente 4,81%. Isso nos possibilitou dizer, com as características de entradas na simulação de portfólios gerados aleatoriamente, que a diferença adotando-se essa aproximação e a por otimização foi muito pequena. É claro que isso dependeu de muitas entradas aqui; como a escolha dos ativos, o período de observação, do intervalo de retorno (diário), do valor de aversão ao risco do investidor, e principalmente do número de portfólios de geração aleatória. Contudo, todos elementos na simulação conseguem carregar discussões interessantes de estatística e programação, e invocar conceitos de matemática financeira.

6

Considerações Finais

Pesquisas como as realizadas pelo [SERASA \(2023\)](#) indicam que a realidade dos brasileiros é o endividamento, o que sugere que o momento é o de gerenciar seus recursos de modo a abandonar a condição de endividado, para passar a condição de poupador e investidor. Neste caso, após sair da condição de endividado sem controle, o ambiente de incertezas de renda variável não é o melhor lugar para alocar recursos inicialmente. Agora, para quem tem recurso e tempo disponíveis, deve ainda estudar bastante antes de se aventurar, pois nesse caso as informações são importantes, podendo ser o diferencial entre ganhar ou perder dinheiro. No entanto, não se pretendeu aqui indicar usar o método apresentado na decisão de escolha de onde investir e quantidades dedicadas a cada um. Objetivamos dar ferramentas computacionais de análise histórica de vários tipos de ativos, fazendo-se uso da estatística descritiva que podem vislumbrar diversas análises para elementos de educação financeira.

[Dias \(2018\)](#) acredita que o Modelo de Markowitz é um importante balizador para a tomada de decisão, porém ele é baseado em uma série histórica e a esperança (ou torcida) de que o comportamento desses ativos se repita no futuro; o que pode ou não ocorrer. Nós indicamos a quem tenha interesse em investir que busque conhecimento de outros modelos de investimentos para julgar o que mais se adequa ao seu perfil, mas, nenhum garante certeza de rendimentos positivos na renda variável.

Aproveitamos o momento para lembrarmos que a aplicação do método neste trabalho não constitui um guia de investimento e sim o de fomentar ideias para quem ensina matemática, em particular o objeto de conhecimento estatística; de modo a aproximar o ensino de matemática ao relevante tema da educação financeira.

O ensino da matemática encontra resistência pelos estudantes, talvez, como conjecturam alguns autores como [Alves \(2020\)](#), porque o problema seja o do ensino da matemática desconectado da realidade. Nosso currículo parece embasar esta decisão quando preconiza que o ensino da matemática deve estar integrado à realidade. Uma outra indicação do currículo e que se faz presente em currículos a partir dos anos 80, é o do uso das tecnologias para o ensino da matemática.

Portanto, identificamos os seguintes problemas: o da má formação dos licenciados em matemática para o ensino da estatística; o do ensino da matemática desconexo da realidade; o da realidade da má gestão dos recursos pessoais dos brasileiros; e o uso da tecnologia para o ensino da matemática. A busca da solução destes problemas motivou a escolha do ensino da estatística, contextualizado por um tema da educação financeira, no caso método de Markowitz para a formação de carteira de investimentos, bem como sua otimização e usando a ferramenta tecnológica do **Google Colaboratory**, permitindo a coleta, limpeza e tratamento de dados.

Após essa contextualização, apresentamos nosso problema de pesquisa: pode o conhecimento de um método de investimento fomentar o ensino de estatística, baseado no modelo de Markowitz e usando tecnologia? Para conseguir responder a esta pergunta nos propomos cumprir o objetivo geral de fomentar o ensino da estatística, na temática da educação financeira e intermediado por tecnologia; e os seguintes objetivos específicos: indicar, por meio de pesquisa bibliográfica em trabalhos e documentos oficiais, a delimitação do objeto do conhecimento, a Estatística; Conhecer o método para formação de carteira de Markowitz, bem como o da otimização da carteira pelo máximo índice Sharpe obtido em pesos gerados aleatoriamente; Apresentar o **colab**, assim como alguns comandos; e, Aplicar o método para uma carteira de investimentos. E agora, já no final do trabalho, chegamos as seguintes conclusões:

Acreditamos ter alcançado o objetivo deste trabalho, pois a utilização do método de Markowitz para a formação de uma carteira de investimentos, nada mais é do que a prática da habilidade:

(EM13MAT202) Planejar e **executar pesquisa amostral** usando dados coletados ou de diferentes fontes sobre **questões relevantes** atuais, incluindo ou não, **apoio de recursos tecnológicos**, e **comunicar os resultados por meio de relatório contendo gráficos e interpretação das medidas de tendência central e das de dispersão**.(BRASIL, 2018, p.546, grifos do autor)

A pesquisa foi realizada tomando em consideração um recorte temporal de um subconjunto de ativos oferecidos na B3 e de criptoativos, logo a ação foi a de **executar pesquisa do tipo amostral**. A questão de investimentos financeiros é **temática relevante** e constitui o currículo sob o tema da educação financeira. Usamos a IDE *Colab* da LP *Python*, que constitui **apoio de recursos tecnológicos**. Os resultados obtidos estão impressos no capítulo 5, que contém **gráficos**; e finalmente o retorno esperado médio e a volatilidade, identificadas por média aritmética e o desvio padrão do conjunto formado pelos elementos que são os preços dos ativos, no fechamento ajustado, tendo conseqüentemente **medidas de tendência central e dispersão**. Portanto, podemos dizer que alcançamos o nosso objetivo.

O alcance desse objetivo só foi possível graças aos objetivos específicos. Com a pes-

quisa documental e bibliográfica sobre o objeto de conhecimento estatística realizada na BNCC e nos trabalhos acadêmicos do Profmat de Alves (2020) e Dias (2018), chegamos nos tópicos de estatística expostos no nosso trabalho: população e amostra; Variáveis; Organização de dados; Medidas de tendencia central e de dispersão; Coeficiente de variação; Covariância e Coeficiente de Correlação. Já sobre o modelo de formação de carteira de investimentos de Markowitz, bem como a idealização de sua otimização, apresentou os seguintes resultados: Retorno; Risco; Diversificação; Fronteira eficiente; Índice de Sharpe; e CML. E por último, a apresentação da IDE *Colab*, que contou com um tutorial de acesso à interface, um conjunto de comandos e apresentação de algumas bibliotecas que posteriormente foram usadas no capítulo 5.

O alcance dos objetivos específicos fica mais visível quando paramos para descrever os passos da execução do método de Markowitz para uma carteira de investimentos: a coleta de dados foi realizada por intermédio da biblioteca *yfinance*, que captura os dados dos ativos pertencentes a B3 e criptomoedas oriundos do site **Yahoo Finance**; a organização em tabelas, por intermédio da biblioteca *Pandas* e a exibição de resultados por meio de gráficos construídos pelas bibliotecas *Matplotlib* e *Plotly*; a biblioteca *Numpy* ficou responsável também por uma boa parte dos cálculos presentes no capítulo 5. Enfim, a descrição do acompanhamento dos passos nos permite perceber o sucesso em cumprir os objetivos específicos supracitados, e servem de guia para que outros professores apresentem suas construções e elementos para análise.

Muito embora tenhamos alcançado nosso objetivo, encontramos alguns problemas; como o de ter que explicar termos como maximização para turmas de ensino médio, o que fica, pra nós, distante da realidade deste público detalhar/ensinar otimização em tão pouco tempo disponível num conteúdo programático extenso de matemática concorrido a de demais disciplinas. Nosso primeiro intuito era o de realizar o cálculo do máximo Índice de Sharpe e o da CVM (mínimo risco) usando apenas a carteira gerada por pesos aleatórios. Porém, ao analisarmos os resultados acabamos fazendo uso de derivadas. O que certamente foge ao escopo do público-alvo, os estudantes do ensino médio. Porém, neste ponto algumas ideias já estavam formalizadas, de modo que, já havíamos criado justificativas para a organização de dados em tabelas e gráficos; e dado uma interpretação para medidas de tendência central como média aritmética, para o cálculo do retorno médio, por exemplo; assim como para medidas de dispersão como desvio padrão, usado para medir o risco (volatilidade). Um outro problema encontrado, foi o de não nos valermos da correlação para a seleção dos ativos, como medida para minimização de risco, problema detectado apenas depois de finalizado o experimento. Muito embora, a seleção acabou parcialmente feita na obtenção da carteira ótima. Acreditamos que todos esses problemas serão temas de trabalho futuro, senão nossos,

de certo de um pesquisador interessado pela temática.

Assim como já mencionado, a relevância do tema se dá por fomentar o ensino da estatística para o ensino médio, que é um objeto de conhecimento onde os professores têm dificuldades formativas; contextualizada por um modelo de investimento, que na prática responde à pergunta, pra que serve isso? Muito ouvida pelos professores, ou seja, dando um significado além dos símbolos matemáticos; e usando a ferramenta tecnológica *Colab*, que nada mais é do que um dos ambientes de desenvolvimento da linguagem de programação *Python*, cujo ensino das noções são previstas em uma única habilidade da BNCC.

Por fim, esperamos ter aguçado a curiosidade de acadêmicos e professores de matemática, que inspirados por este e outros textos, sintam-se no desafio e estímulo de ensinar por meio de temas relevantes, como são os da matemática financeira! Isso, pensamos nós, ser um passo para auxiliar seus alunos a mudar a realidade da má gestão de recursos financeiros.

Referências

- ALVES, D. P. *A Estatística como auxílio na seleção de carteiras de investimento: uma aplicação por meio do modelo de Markowitz*. Dissertação (Mestrado) — Universidade de Brasília – ProfMat, 2020. [8](#), [11](#), [12](#), [22](#), [23](#), [24](#), [38](#), [39](#), [40](#), [42](#), [44](#), [47](#), [105](#), [107](#)
- BACCAR, M. H. M. M.; VELASQUE, L. de S.; LEAL, V. M. Pesquisando saberes estatísticos trazidos da educação básica em licenciandos de matemática. In: SIMAS, F.; CAMBRAINHA, M. (Ed.). *Anais do 6º Simpósio Nacional da Formação de Professores de Matemática*. Rio de Janeiro, RJ: ANPMat, 2023. v. 6, n. 6, p. 242–249. [12](#), [13](#), [17](#)
- BODIE, Z.; KANE, A.; MARCUS, A. *Fundamentos de investimentos*. Porto Alegre, RS: AMGH Editora, 2010. [44](#), [48](#)
- BORGES, L. E. *Python para desenvolvedores*. São Paulo, SP: Novatec Editora, 2014. [53](#), [57](#), [58](#), [59](#), [62](#)
- BRASIL. *Parâmetros Curriculares Nacionais Ensino Fundamental*. 1998. [16](#)
- BRASIL. *Parâmetros Curriculares Nacionais Ensino Médio*. 2000. [16](#)
- BRASIL. *Plano-Diretor-ENEF-Estrategia-Nacional-de-Educacao-Financeira*. 2010. [14](#)
- BRASIL. *Base Nacional Comum Curricular*. 2018. [15](#), [16](#), [17](#), [20](#), [21](#), [23](#)
- BUSSAB, W. de O.; MORETTIN, P. A. *Estatística básica*. São Paulo: Saraiva Uni, 2017. [30](#), [32](#), [34](#)
- CAMPOS, S. T. de A. P. *Matemática financeira no ensino médio: uma proposta de ensino contextualizada, utilizando planilhas eletrônicas*. Dissertação (Mestrado) — Universidade Federal de São Paulo – ProfMat, 2021. [14](#), [15](#), [17](#)
- CAZORLA, I. M.; UTSUMI, M. C.; MONTEIRO, C. E. F. Reflexões sobre as variáveis estatísticas e suas representações em gráficos. *Investigações Hispano-Brasileiras em Educação Estatística*, p. 189–195, 2020. [26](#), [27](#)
- DIAS, D. P. *Teoria de Markowitz e programação linear para formação de uma carteira ótima de investimentos*. Dissertação (Mestrado) — Universidade Federal de São Carlos, 2018. [8](#), [11](#), [22](#), [23](#), [30](#), [32](#), [34](#), [38](#), [40](#), [41](#), [42](#), [44](#), [47](#), [105](#), [107](#)
- FRANZONI, P.; QUARTIERI, M. T. Tarefas investigativas relacionadas à educação financeira: possibilidades de conjecturas e estratégias de resolução. *Ciência & Educação (Bauru)*, SciELO Brasil, v. 26, 2020. [13](#), [15](#), [16](#)

GALDINO, S. C. de O.; RODRIGUES, F. L. Aplicação do modelo de Markowitz e do índice de Sharpe na otimização de um portfólio de risco. *Revista Conhecimento Contábil*, v. 12, n. 1, 2022. 39, 42, 43, 44, 46, 47, 48, 49, 50, 51

GIORDANO, C.; ALVES, J.; QUEIROZ, C. de. Educação estatística e a base nacional comum curricular: o incentivo aos projetos. *REVEMAT: Revista Eletrônica de matemática*, Universidade do Extremo Sul Catarinense, v. 14, p. 1–20, 2019. 13, 17

GIORDANO, L. V. O. *Panorama da competência estatística no ensino médio brasileiro= das ideias e práticas dos professores ao desempenho dos alunos no ENEM*. Tese (Doutorado) — Universidade de Campinas, 2017. 11, 12, 13, 17, 19, 20

HIEDA, A.; ODA, A. L. Um estudo sobre a utilização de dados históricos no modelo de Markowitz aplicado a Bolsa de Valores de São Paulo. *Seminários de Administração*, v. 3, p. 1998, 1998. 41, 44, 45, 47, 50

IEZZI, G.; HAZZAN, S.; DEGENSZAJN, D. M. *Fundamentos de matemática elementar, 11: matemática comercial, matemática financeira e estatística descritiva*. São Paulo, SP: Atual Editora, 2013. 19, 24, 30, 32, 38, 39

KUNTZ, E. R. *A Matemática Financeira no Ensino Médio como fator de fomento da educação financeira: resolução de problemas e letramento financeiro em um contexto crítico*. Dissertação (Mestrado) — Pontifícia Universidade Católica de São Paulo, 2019. 14

LEITE, G. N. *Python e Fortran como ferramenta na resolução de problemas de matemática e física*. Dissertação (Mestrado) — Universidade Federal do Ceará – ProfMat, 2020. 52

LIMA, L. L. B. dos S. *Seleção de carteiras ótimas pelo método de Markowitz: Análise do Índice de Sharpe, Índice de Treynor e Alfa de Jensen no Microsoft Excel*. Dissertação (TCC de Graduação) — Universidade Federal da Paraíba, 2019. 39, 49

LIMA, T. C. de S. *Aplicação do modelo de Markowitz para a otimização de carteiras de títulos públicos*. Dissertação (TCC de Graduação) — Universidade Tecnológica Federal do Paraná, 2016. 38, 39, 41, 42

MAGALHÃES, M. N.; LIMA, A. C. P. de. *Noções de probabilidade e estatística*. São Paulo, SP: Editora da Universidade de São Paulo, 2023. v. 7. 11, 19, 23, 24, 26, 28, 30, 32, 34

MARKOWITZ, H. M. Foundations of portfolio theory. *The Journal of Finance*, [American Finance Association, Wiley], v. 46, n. 2, p. 469–477, 1991. ISSN 00221082, 15406261. Disponível em: <<http://www.jstor.org/stable/2328831>>. 39, 41

MARTINS, M. O.; LENHARD, C. Classificador para predição de aluno evasor de graduação em universidades particulares. *Disciplinarum Scientia – Ciências Naturais e Tecnológicas*, v. 22, n. 2, p. 105–118, 2021. 66

MENEZES, N. N. C. *Introdução à programação com Python*. São Paulo, SP: Novatec Editora, 2019. 52, 53, 57, 58, 62, 65

NEVES, L. L. G.; COSTA, R. R. Machine Learning para criação de carteira eficiente de Markowitz. In: *Engenharia, Gestão e Inovação*. Belo Horizonte, MG: Editora Poisson, 2022. v. 6, p. 58–70. 65, 66

PAIVA, F. A. P. et al. *Introdução a Python com aplicações de sistemas operacionais*. Natal, RN: Editora IFRN, 2021. 62

PESSOA, M. M. *Proposta de atividades de matemática financeira para o Ensino Médio nos termos da Nova Base Nacional Comum*. Dissertação (Mestrado) — Universidade Federal do Espírito Santos – ProfMat, 2022. 14, 15, 17

RIBEIRO, C. T. Agenda em políticas públicas: a estratégia de educação financeira no Brasil à luz do modelo de múltiplos fluxos. *Cadernos EBAPE. BR*, SciELO Brasil, v. 18, p. 486–497, 2020. 13, 15, 16

ROSS, S. et al. *Fundamentos de administração financeira*. Porto Alegre, RS: Bookman Editora, 2022. 40, 41, 42

SCHIROKY, M. M. *Seleção de carteira através do modelo de Markowitz*. Dissertação (Especialização) — Universidade Federal do Rio Grande Sul, 2007. 41, 44, 48, 49, 50

SILVA, J. M. N. *Educação financeira e matemática financeira na BNCC: percepções de professores que ensinam matemática na educação básica*. Dissertação (Mestrado) — Universidade do Estado do Mato Grosso, 2021. 14, 52

SODRÉ, A. A. S. L. *Investigando uma sequência didática sobre juros compostos para a formação em educação financeira de alunos do ensino médio*. Dissertação (Mestrado) — Universidade Federal de Ouro Preto, 2018. 14, 17

WAZLAWICK, R. S. *Introdução a Algoritmos e Programação com Python: Uma abordagem dirigida por testes*. Rio de Janeiro: Elsevier Editora Ltda., 2018. 52

YAMAMOTO, R. A. *Análise computacional do risco X rendimento de um portfólio de investimento a partir do cálculo da fronteira de Markowitz*. Dissertação (Trabalho de Conclusão de Graduação) — Universidade Estadual Paulista (Unesp), 2022. 65, 66

O Banco Paulista, em atendimento às exigências regulatórias, elaborou o Questionário API com objetivo de auxiliá-lo a identificar seu perfil de investidor (**conservador, moderado ou arrojado**) e recomendar os investimentos mais alinhados a esse perfil. Para cada questão, escolha a alternativa que mais se aproxima dos seus objetivos e expectativas de investimento.

QUESTIONÁRIO PARA ANÁLISE DE PERFIL DO INVESTIDOR

1. Por quanto tempo você pretende deixar seu dinheiro investido?

- a Menos de 6 meses
- b Entre 6 meses e 1 ano
- c Entre 1 ano e 3 anos
- d Acima de 3 anos

2. Qual o objetivo desse investimento?

- a Preservação do capital para não perder valor ao longo do tempo, assumindo baixos riscos de perdas
- b Aumento gradual do capital ao longo do tempo, assumindo riscos moderados
- c Aumento do capital acima da taxa de retorno média do mercado, mesmo que isso implique assumir riscos de perdas elevadas
- d Obter no curto prazo retornos elevados e significativamente acima da taxa de retorno média do mercado, assumindo riscos elevados

3. Qual das alternativas melhor classifica sua formação e experiência com o mercado financeiro?

- a Não possuo formação acadêmica ou conhecimento do mercado financeiro
- b Possuo formação acadêmica na área financeira, mas não tenho experiência com o mercado financeiro
- c Possuo formação acadêmica em outra área, mas possuo conhecimento do mercado financeiro
- d Possuo formação acadêmica na área financeira ou pleno conhecimento do mercado financeiro

4. Considerando seus rendimentos regulares, qual a porcentagem você pretende reservar para aplicações financeiras?

- a No máximo 25%
- b Entre 25,01 e 50%
- c Acima de 50%

5. Caso as suas aplicações sofressem uma queda superior a 30%, o que você faria?

- a Resgataria toda a aplicação e aplicaria na poupança
- b Manteria aplicação aguardando uma melhora do mercado
- c Aumentaria a aplicação para aproveitar as oportunidades do mercado

6. Como está distribuído o seu patrimônio?

- a Meu patrimônio não está aplicado ou está todo aplicado em renda fixa e/ou imóveis
- b Menos de 25% em renda variável e o restante em renda fixa e/ou imóveis
- c Entre 25,01 e 50% aplicado em renda variável e o restante em renda fixa e/ou imóveis
- d Acima de 50% em renda variável

7. Em relação as aplicações e rendimentos, em qual dessas situações você se enquadra?

- a Conto com o rendimento dessas aplicações para complementar minha renda mensal
- b Eventualmente posso resgatar parte das aplicações para fazer frente aos meus gastos. Contudo, não tenho intenção de resgatar no curto prazo e pretendo fazer aplicações regulares
- c Não tenho intenção de resgatar no curto prazo, mas não pretendo realizar novas aplicações
- d Não tenho intenção de resgatar no curto prazo e ainda pretendo fazer aplicações regulares

8. Indique em quais aplicações listadas abaixo você já investiu e qual a frequência nos últimos dois anos. Pode assinalar mais do que uma alternativa.

Tipos de Aplicações	Nunca investiu	1 a 2 vezes	3 ou + vezes
<ul style="list-style-type: none"> » Fundos Renda Fixa Simples ou Indexados » Fundos Renda Fixa Duração Baixa, Média ou Livre » Renda Fixa Títulos Públicos » Renda Fixa (CDB, LCI, LCA, LF Sênior) indexada ao CDI » Fundos Multimercados Balanceados, Dinâmicos ou de Capital Protegido » Títulos Não Financeiros (Debêntures, CRI, CRA, CDCA, CCB, CPR) indexado ao CDI – Emissor com Grau de Investimento » LF Subordinada – Emissor com Grau de Investimento » BTC – Posições doadoras em Empréstimos de Ações 	0	1	1
<ul style="list-style-type: none"> » FIDC » Títulos Não Financeiros (Debêntures, CRI, CRA, CDCA, CCB, CPR) indexado ao CDI – Emissor sem Grau de Investimento 	0	2	3
<ul style="list-style-type: none"> » Fundos Renda Fixa Duração Média ou Livre – Crédito Livre » Fundos Renda Fixa Duração Alta » Fundos Ações » Fundos com Alavancagem » Fundos Multimercados » Fundos Imobiliários » FIP » FIDC – Outros Indexadores ex CDI – Emissão sem Grau de Investimento – Prazo acima de 3 anos » Fundo Cambial » Ações (Mercado à vista, BDR, ETF) » LF Subordinada – Emissor sem Grau de Investimento » BTC – Posições tomadoras em Empréstimos de Ações » Derivativos Listados em Bolsa (Futuros, Opções e Termos) 	0	5	6

Obs: Efetue a somatória dos pontos referente a questão acima e escolha a alternativa equivalente:

- a Entre 0 e 1 ponto
- b Entre 2 e 4 pontos
- c Acima de 5 pontos

9. Tomando por base as respostas da questão anterior, informe o volume aproximado que foi destinado a cada operação no período.

Tipos de Aplicações	Nunca investiu	Menos de R\$ 10mil	Entre R\$ 10.001 a R\$ 50.000	Entre R\$ 50.001 a R\$ 100.000	Acima de R\$ 100.001
<ul style="list-style-type: none"> » Fundos Renda Fixa Simples ou Indexados » Fundos Renda Fixa Duração Baixa, Média ou Livre » Renda Fixa Títulos Públicos » Renda Fixa (CDB, LCI, LCA, LF Sênior) indexada ao CDI » Fundos Multimercados Balanceados, Dinâmicos ou de Capital Protegido » Títulos Não Financeiros (Debêntures, CRI, CRA, CDCA, CCB, CPR) indexado ao CDI – Emissor com Grau de Investimento » LF Subordinada – Emissor com Grau de Investimento » BTC – Posições doadoras em Empréstimos de Ações 	0	1	1	1	1
<ul style="list-style-type: none"> » FIDC » Títulos Não Financeiros (Debêntures, CRI, CRA, CDCA, CCB, CPR) indexado ao CDI – Emissor sem Grau de Investimento 	0	2	3	4	5
<ul style="list-style-type: none"> » Fundos Renda Fixa Duração Média ou Livre – Crédito Livre » Fundos Renda Fixa Duração Alta » Fundos Ações » Fundos com Alavancagem » Fundos Multimercados » Fundos Imobiliários » FIP » FIDC – Outros Indexadores ex CDI – Emissão sem Grau de Investimento – Prazo acima de 3 anos » Fundo Cambial » Ações (Mercado à vista, BDR, ETF) » LF Subordinada – Emissor sem Grau de Investimento » BTC – Posições tomadoras em Empréstimos de Ações » Derivativos Listados em Bolsa (Futuros, Opções e Termos) 	0	7	8	9	10

Obs: Efetue a somatória dos pontos referente a questão acima e escolha a alternativa equivalente:

- a 1 ponto
- b Entre 2 e 6 pontos
- c Acima de 7 pontos