Universidade de Brasília
Faculdade de Economia, Administração, Contabilidade e
Gestão de Políticas Públicas
Departamento de Administração

THIAGO FELIX DE CARVALHO

Análise exploratória de dados da ANAC:
Uma perspectiva sob a linguagem de programação Python

Brasília - DF
2022

THIAGO FELIX DE CARVALHO

Análise exploratória de dados da ANAC:
Uma perspectiva sob a linguagem de programação Python

"Monografia apresentada ao Departamento de Administração como requisito parcial à obtenção do título de Bacharel em Administração."

Professor Dr. Orientador:
Victor Rafael Rezende Celestino

Brasília – DF
2022

THIAGO FELIX DE CARVALHO

Análise exploratória de dados da ANAC:
Uma perspectiva sob a linguagem de programação Python

A Comissão Examinadora, abaixo identificada, aprova o Trabalho de Conclusão do Curso de Administração da Universidade de Brasília do (a) aluno (a)

Thiago Felix de Carvalho

Professor Dr., Victor Rafael Resende Celestino
Professor-Orientador

Professora Dra. Silvia Araújo dos Reis,
Professora-Examinadora

Professor Dr. Carlos Rosano Peña
Professor-Examinador

Brasília, de setembro de 2022.

Dedico especialmente ao meu Professor Orientador, Dr. Victor Rafael Rezende Celestino, que sempre demonstrou paciência infinita, e sempre a alegria no olhar de quem é apaixonado pelo tema da aviação civil. De tal modo que fui levado pelas mesmas veredas, e de tudo que formular, o terei sempre em vista.

AGRADECIMENTOS

Agradeço primeiramente a minha mãe, que me deu a vida. E isso é tudo o que eu poderia receber dela. Cada contexto social nos oferece a possibilidade de recebermos, além da vida, os fatores que a compõem naquele cenário, e que mudam com uma constância sem fim. A vida não. Esta permanece estável. Além disso, a força que presenciei de tal criatura é tamanha, que nela encontro o poder de Deus. Nada mais é relevante.

E com isso se faz tudo!

“A real contribuição da estatística para a sociedade é principalmente moral, não técnica.”

Max Morris

RESUMO

Esta pesquisa tem como proposta oferecer uma releitura da análise exploratória de dados (EDA) da Agência Nacional de Aviação Civil (ANAC), utilizando a linguagem de programação Python. Atualmente a Agência utiliza o Microsoft Power Business Inteligence para apresentar informações referentes ao transporte aéreo de passageiros no Brasil. Preços, quantidade de passageiros e voos, índices macroeconômicos, entre outras informações, são oferecidas graficamente. Sabe-se que a linguagem de programação Python permite uma maior otimização da EDA e leituras gráficas, permitindo maiores configurações e uma análise mais voltada aos interesses do usuário, facilitando sua apresentação, por conseguinte sua análise. Deste modo, esta pesquisa tem como objetivo reavaliar a atual apresentação de dados do site da agência, a fim de propor melhorias para que a sociedade civil, principal influenciado pelas informações em questão, possa compreender com facilidade o que significam tais informações. A metodologia utilizada é a de *Design Science.* Logo, esta pesquisa pretende obedecer às 12 etapas desta aplicação metodológica. Os resultados apresentam informações remodeladas, compreendendo a apresentação de dados como um storytelling, de modo que se contextualize a sua razão de ser.
Palavras-chave: análise de dados, python, aviação civil, sociedade civil, storytelling.

ABSTRACT

This research aims at offering a reinterpretation of the data modeling used by the National Civil Aviation Agency (ANAC), using the Python programming language. Currently, the Agency uses Microsoft Power Business Intelligence to present information regarding air transport of passengers in Brazil. Prices, number of passengers and flights, macroeconomic indices, among other information, are provided graphically. It is known that the Python programming language allows a greater optimization of the modeling and graphic reading, allowing greater configurations and a modeling more focused on the user's interests, facilitating its presentation, therefore, its analysis. Thus, this resarch aims at reassess the current presentation of data on the agency's website, in order to propose improvements so that civil society, mainly influenced by the information in question, can easily understand what such information means. The methodology used is Design Science. Therefore, this research intends to obey the 12 steps of this methodological application. The results presents remodeled information, comprising the presentation of data as a storytelling, so that its reason for being is contextualized.
Keywords: data analysis, python, civil aviation, civil society, storytelling.

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

A Agência Nacional de Aviação Civil (ANAC) é uma agência reguladora federal que regula e fiscaliza as atividades da aviação civil e a infraestrutura aeronáutica e aeroportuária no Brasil. Criada em 2005, passou a atuar a partir de 2006, em substituição ao Departamento de Aviação Civil (DAC). Possui regime especial, vinculada ao Ministério da Infraestrutura. Suas ações se enquadram em macroprocessos de certificação, fiscalização, normatização e representação institucional.

Sua missão é garantir a segurança e a excelência da aviação civil. A agência tem como visão ser referência na promoção da segurança e no desenvolvimento da aviação civil, entre as quais se destacam os seguinte valores:

- Segurança é o nosso propósito.
- Atuamos com foco no resultado e no interesse público.
- Trabalhamos com autonomia e competência técnica.
- Agimos com integridade, comprometimento e transparência.
- Valorizamos as pessoas e suas competências.
- Incentivamos a inovação e a cooperação no setor de aviação civil.
- Temos orgulho de trabalhar na ANAC.

Suas competências foram reguladas pela Lei nº 11.182, de 27 de setembro de 2005 – Lei de criação da ANAC, segunda a qual estabelece que a Agência regula e fiscaliza as atividades da aviação civil e da infraestrutura aeronáutica e aeroportuária, observadas as orientações, políticas e diretrizes do Governo Federal. Dentre as principais competências destacam-se:

- ✓ "Representar o Brasil junto a organismos internacionais de aviação e negociar acordos e tratados sobre transporte aéreo internacional."
- ✓ "Emitir regras sobre segurança em área aeroportuária e a bordo de aeronaves civis."
- ✓ "Conceder, permitir ou autorizar a exploração de serviços aéreos e de infraestrutura aeroportuária."
- ✓ Estabelecer o regime tarifário da exploração da infraestrutura aeroportuária.
- ✓ Administrar o Registro Aeronáutico Brasileiro (RAB).
- ✓ Homologar, registar e cadastrar os aeródromos.
- ✓ "Emitir certificados de aeronavegabilidade atestando aeronaves, produtos e processos aeronáuticos e oficinas de manutenção."
- ✓ Fiscalizar serviços aéreos e aeronaves civis.

- ✓ Certificar licenças e habilitações dos profissionais de aviação civil.
- ✓ "Autorizar, regular e fiscalizar atividades de aeroclubes e escolas e cursos de aviação civil."
- ✓ "Reprimir infrações às normas do setor, inclusive quanto aos direitos dos usuários, aplicando as sanções cabíveis."

Cabe a agência trabalhar para promover a segurança da aviação civil e para estimular a concorrência e a melhoria da prestação dos serviços no setor. Isto consiste em elaborar normas e procedimentos, certificar empresas do setor, oficinas, escolas de formação, profissionais da aviação, aeródromos e aeroportos, além de fiscalizar as operações de aeronaves, de empresas aéreas, de aeroportos e de profissionais do setor, com objetivo de garantir segurança e qualidade no transporte aéreo.

Dentro deste escopo, cabe à ANAC revisar, atualizar e editar regulamentos técnicos e relacionados a aspectos econômicos. Essas normas são criadas a partir de audiências publicas anteriores a sua elaboração, a fim de ouvir a sociedade, e de estudos sobre o potencial de cada decisão da agência. Suas normas estão em acordo com as principais organizações internacionais de aviação das quais o Brasil é signatário.

As certificações pelas quais a ANAC é responsável visam atestar o grau de confiança e o atendimento a requisitos do regulamento internacional da aviação. Ela certifica aeronaves e helicópteros, bem como seus componentes, oficinas de manutenção, empresas do setor, escolas e profissionais da aviação.

Para realizar o funcionamento da aviação e garantir níveis de qualidade e segurança na prestação dos serviços aos passageiros, a agência realiza atividade de vigilância continuada, além de ações fiscais. Na vigilância, o acompanhamento sobre o desempenho de produtos, empresas, operações, processos e serviços e dos profissionais certificados se dá de maneira planejada e contínua.

Quanto às autorizações e concessões para atuar, companhias, empresas de táxi-aéreo ou de serviços especializados, oficinas de manutenção, escolas, operadores de aeródromos e profissionais da aviação precisam ser autorizados pela ANAC. De acordo com a complexidade de cada atividade, a agência emite autorizações, permissões, outorgas ou concessões a esses entes regulados por ela. Qualquer infração ou descumprimento pode levar à cassação da concessão concedida.

Além dos setores, diversas categorias profissionais são necessárias para que o transporte aéreo aconteça de forma organizada. Pilotos, comissários de bordo, despachantes operacionais de voos, mecânicos de manutenção, agentes de proteção à aviação civil e bombeiros de

aeródromos são alguns dos exemplos. Compete a agência emitir licenças e certificados de habilitações técnicas par que esses profissionais possam atuar na aviação civil.

A instrução normativa nº 127, de 5 de outubro de 2018, determina regras e diretrizes para edição do regimento interno, organização das unidades organizacionais e para os processos de modificação da estrutura organizacional da agência, de acordo com o site a reguladora do transporte aéreo.

Recentemente a ANAC lançou uma plataforma desenvolvida em Power BI (*bussiness inteligence)*, ferramenta da Microsoft que coleta dados e informações para tratar, realizar cálculos a fim de criar indicadores para oferecer uma tomada de decisão mais acertada. Essa ferramente reúne informações de diversas fontes, para apresentá-los de maneira visualmente resumida e ajustada para facilitar a identificação de *insights.* Um dos benefícios desse instrumento é a possibilidade de trabalhar com milhares de informações sem travas, otimiza a realização de análises, que muitas vezes ferramentas comuns não são capazes de fazer, como o Microsoft Excel.

A ferramenta permite analisar indicadores de tarifas aéreas comercializados em um painel interativo, facilitando a compreensão de diferentes públicos. É possível visualizar preço médio do bilhete aéreo e do quilômetro pago por Unidade da Federação, ranking de tarifas comercializada por UF, proporção de bilhetes vendidos por faixa de preço, valor da tarifa e variação por empresa, além da comparação entre UF por período, segundo o site Passageiro de Primeira (2022).

Além disso, indicadores macroeconômicos também serão mostrados, como a cotação do dólar, valor da taxa Selic, preço médio do barril de petróleo e a inflação aferida para o período medida pelo Índice de Preços ao Consumidor Amplo (IPCA). Para todos eles é possível recortar por período analisado. Abaixo é possível verificar como as tarifas são apresentadas:

Figura 1 - Tarifas da aviação doméstica

Fonte: ANAC, 2022. Tarifas da aviação doméstica.

Além deste, é apresentado ainda 6 painéis com os dados descritos acima. Apesar se ser muito mais fluido a leitura dos dados atuais, a ferramenta ainda apresenta algumas falhas que impedem uma utilização mais intuitiva, como a colocada na imagem abaixo, onde os valores de 2021 referente às tarifas desaparece ao percorrer o mouse pelo painel interativo.

Figura 2 - Aba 7/8 – Tarifas da aviação doméstica

Fonte: ANAC, 2022

1.2 Formulação do problema

Antes dessa evolução era sabido que a forma como os dados eram expostos no portal da ANAC não pareciam ser visualmente favoráveis para qualquer análise, e que muitas vezes o sujeito que necessitava delas tinha de se debruçar por uma série de planilhas, muitas vezes não tão claras, para que se pudesse perceber um cenário relacionado ao que se pretender compreender no que se refere às tarifas praticadas pelo setor de aviação.

Por outro lado, ainda se tem que, para se fazer uma análise exploratória dos dados disponíveis se levaria tempo e muito esforço até descobrir uma forma de utilizá-los para se tomar uma decisão ou fazer qualquer estudo. Logo, a questão desta pesquisa é “Como analisar e relatar uma quantidade muito grande de dados, em uma linguagem clara, que permita a compreensão do significado das informações nesses contidas?”.

1.3 Objetivo Geral

O objetivo deste trabalho é compor um instrumento de extração e leitura de base de dados, de forma que estas sejam analisadas e relatadas em uma linguagem visual clara, por meio de gráficos que possam exprimir o real significado de uma quantidade muito grande de informações.

A necessidade de se criar uma ferramenta para extrair, analisar e projetar esses dados, a fim de apresentar um cenário que seja legível e compreensível para qualquer um se torna imprescindível para que a informação seja útil para o maior número de pessoas possível. Esta alteração considerável na leitura facilitou bastante sua análise, embora ainda apresente falhas visuais que veremos mais adiante.

1.4 Objetivos Específicos

Pretende-se alcançar três principais objetivos, entre eles:

1. Criar um modelo de Notebook em Python que extraia os dados do portal ANAC e gere um modelo gráfico de fácil compreensão;
2. Criar um Notebook que analise os dados extraídos;
3. Apresentar graficamente os resultados.

1.5 Justificativa

Sabe-se da capacidade que a linguagem de programação em Python tem de realizar personalizações detalhadas de análise de dados, que fogem da alçada do BI (business inteligente). Dessa forma, pretende-se mostrar como essa integração, e algumas vezes, o uso específico desta linguagem pode otimizar a apresentação dos dados analisados pela plataforma lançada pela agência para analisar os dados coletados do setor aéreo.

Assim, espera-se que este trabalho contribua para uma melhor apresentação e leitura, de forma que o conjunto de personalizações dos dados analisados sejam maiores que aquelas disponíveis com a ferramenta Microsoft Power Bussiness Inteligence.

Sabe-se ainda que a maior parte da população não detém domínio para coleta, processamento, e geração de relatórios claro e concisos que as quantidades de informações prestadas pela agência exigem. Por isso, este trabalho tem como interesse oferecer uma ferramenta que acesse a quantidade de dados oferecidas, e crie uma apresentação visualmente mais atrativa para mostrar as estatísticas do setor, a fim de esclarecer possíveis interessados. Ao mesmo tempo em que o interesse da sociedade pelas informações prestadas pela agência pode estar atrelado à facilidade com que esses dados são dispostos, de forma a facilitar a compreensão, por se tratar de uma grande quantidade de dados e que simplesmente não podem ser analisados para se tirar conclusões apenas os visualizando da maneira com eram oferecidos.

Por se tratar de uma pesquisa de interesse público, é mister posicionar a Agência Reguladora como um dos principais StakeHolders desta pesquisa. Só o fato de falarmos sobre uma ferramenta que pode contribuir sobremaneira na forma como a agência posiciona suas

informações coletadas do setor, a aplicá-la em seu site público, já a coloca não apenas como uma interessada, mas como a principal parceira desta relevante pesquisa. Ainda se tem o fato de usuários de uma maneira geral, programadores, gestores, empresários do setor ou mesmo companhias aéreas interessadas, se interessarem pela ferramenta a fim de aplicarem em seus processos digitais de forma a obter acesso facilitado aos números, que possuem relevância extrema, para tomada de decisão em suas estratégias de mercado.

Por último, justifica-se pela importância acadêmica, tal como um acréscimo nos estudos constantes do setor aéreo, de modo a oferecer ainda mais possibilidades de gestão de dados digitais, processamento, EDA e aplicação, oferecendo aperfeiçoamento contínuo, dado seu modelo de uso aplicado a esta pesquisa. Bem como contribuirá para a construção deste Trabalho de Conclusão de Curso, capacitando mais um profissional para o mercado de trabalho. Bem como o tema sobre o qual é discutido aqui está em franca ascensão no mercado de trabalho, possibilitando novos estudos na aréa de análise de dados.

2 REVISÃO TEÓRICA

Nesta revisão, a linha de raciocínio estatístico do autor citado foi utilizada. Por isso, não é incomum encontrar, no discorrer desta etapa da revisão, uma estrutura lógica semelhante à do livro de Anderson et al. (2021).

É oportuno ainda frisar que esta etapa da revisão da literatura corresponde à terceira etapa do *DS,* método utilizado por este trabalho. Por isso, na listagem dos doze métodos do *DS,* esta etapa será ocultada, pois será desenvolvida aqui neste item da pesquisa.

Por se tratar de um trabalho com fundamento na estatística descritiva, nenhum outro caminho seria mais oportuno de revisar senão o dos conceitos fundamentais dessa ciência. A estatística está presente no contexto atual da sociedade da mesma maneira como o ar que respiramos. Dito de outra forma, a estatística compõe uma série de ferramentas para tomada de decisão das empresas em todos os setores, e no contexto a que mais interessa este trabalho, nas áreas de administração e economia.

A estatística tem como premissa a obtenção de resultados a partir de valores numéricos, entre os quais estão os percentuais, médias, mínimos e máximos, valores numéricos diversos, medianas, modas, e que compõem uma série de resultados esclarecedores sobre um conjunto de dados analisados. Esses dados podem ser da ordem de dezenas, centenas, milhares ou mesmo bilhões de valores. Os resultados acima poderão oferecer uma série de *insights* significativos para tomada de decisão no âmbito empresarial, seja para a área de recursos humanos, ou mesmo para o setor financeiro.

Ocorre que, muitas vezes, as análises de dados em grande escala podem favorecer sobremaneira a forma como analisamos um ambiente, estatisticamente. Eles poderão oferecer uma visão mais esclarecida do conjunto de fatores que determinam a realidade das organizações. Muitas vezes, este é o único recurso que nos serve, efetivamente.

Esta pesquisa visa apresentar dados, que segundo Anderson (et. al. 2021), são fatos e números coletados, analisados e sintetizados para apresentação e interpretação. Assim, o leitor obtém um resumo claro do que foi analisado, estudado, esmiuçado por dias, semanas ou meses.

Os dados podem se apresentados de diversas maneiras, entre as quais estão os gráficos e tabelas. Estas fazem parte de uma variedade de ferramentas de apresentação de dados. Apresentá-los é uma verdadeira arte, como afirma Knaflic (2019). Apenas muita prática para fazê-lo com eficácia. E este é um importante passo do processo, sobre o qual recai a maior parte do esforço em manipulá-los, a fim de mostrá-los com clareza.

Os dados possuem duas classificações principais, entre elas estão os dados categorizados e os dados quantitativos. Como o próprio nome diz, dados categorizados são os que estão passíveis de categorização, que recebem um agrupamento por classificação categórica. Enquanto dados quantitativos são dados valores numéricos, ou que podem ser dimensionados por escala de razão ou escala intervalar, conforme aponta Anderson et al. (2021).

Assim como levanta o autor, a importância do dimensionamento de custos de operação para se levantar dados estatísticos deve ser analisada. Assim como podem existir fontes de dados quando se tem pouco tempo para se realizar uma pesquisa, outras vezes se faz necessário a obtenção dessas fontes por meio de pesquisa, o que envolve mais tempo e recursos. Portanto, é importante identificar as vantagens de se utilizar de fontes já existentes, bem como a importância e o impacto da pesquisa tanto na tomada de decisão, como também para se criar uma oferta de novo produto ou serviço, por exemplo. Portanto uma ferramenta que possa fazer o levantamento a partir de fontes de dados já existentes, a fim de acelerar o processo de obtenção da informação requerida, para tomar decisões ágeis no ambiente corporativo, pode ser ímpar para o sucesso de um novo projeto, ou mesmo para acompanhar as tendências de mercado.

Para isso, a estatística descritiva fornece os instrumentos adequados de leitura de dados, oferecendo por meio de gráficos e planilhas um resumo numérico de tais fontes. Quando tratamos de grandes dados, denominamos a esta prática o termo de *Analytics*, que é quando se transforma dados em conceitos palpáveis, definições, conclusões e análises numéricas sobre a realidade analisada.

Neste contexto, para que se possa sintetizar os dados de variáveis quantitativas, se utiliza a distribuição de frequência, que nada mais é que um resumo tabular de dados mostrando o número/frequência de observações em cada uma das diversas classes não sobrepostas, conforme aborda Anderson (et. al. 2021).

De acordo com o autor, existem três etapas para definir as classes de uma distribuição de frequências com dados quantitativos. São elas:

1. Determinar o número de classes não sobrepostas.
2. Determinar a amplitude de cada classe.
3. Determinar os limites da classe.

As Classes são formadas delimitando-se os intervalos para agrupar os dados. A recomendação é utilizar entre 5 e 20 classes. O principal objetivo nesta etapa é não criar classes que possuam poucos dados, nem poucas de modo que não se mostrem quantitativamente classificados.

Quanto a amplitude, a prática é que esta seja igual para todas as classes, ao mesmo tempo em que esta não é uma decisão discricionária, mas tendo como base o maior e o menor valor dos dados. Em seguida, é possível obter a amplitude da classe utilizando a seguinte fórmula:

$$\textbf{Amplitude da classe aproximada} = \frac{\textbf{Maior valor dos dados – Menor valor dos dados}}{\textbf{Números de Classes}} = x$$

A depender de quais dados estão sendo analisados, pode-se arredondar o valor atribuído à amplitude da classe. Se está se analisando pessoas, por exemplo, poderia se arredondar o valor hipotético de 6,28 para 7. E assim por diante.

Por último, para os limites de classe é preciso definir a que classe cada dado número pertence, de modo que em cada uma coexista um *limite de classe inferior*, e um *limite de classe superior*. Feito isso, um determinado dado não pode pertencer a duas classes distintas, mas exclusivamente a uma delas, respeitando seus limites.

A distribuição de frequência é o primeiro passo e mais importante para organização dos dados, pois ela mostra como e com que frequência esses dados aparecem, facilitando a visualização, algo que não ocorre quando esses estão desorganizados.

2.1 Visualização de dados

Segundo Anderson (2021), a visualização de dados é um termo empregado para descrever o uso de representações gráficas a fim de sintetizar e apresentar informações sobre um determinado conjunto de dados. O seu objetivo é apresentar de maneira clara e sucinta o que representa os dados analisados. Como argumenta Knaflic (2019), essa representação gráfica requer um tempo precioso analisando os dados, e identificando o que se deseja passar ao seu ouvinte. Ainda argumenta que a melhor forma de apresentação é evitar saturações, e apresentar somente o essencial, focalizando os pontos se que deseja apresentar. Em seu livro *Storytelling com Dados* (KNAFLIC, 2019), resume em seis tópicos quais os principais princípios da apresentação de dados usando storytelling. São eles:

1. Entender o contexto
2. Escolher um visual apropriado
3. Eliminar a saturação
4. Chamar a atenção para onde você quer
5. Pensar como um designer
6. Contar uma história

A primeira premissa para apresentação de dados é ter certeza de que de se tem um sólido conhecimento do que se pretende apresentar ou comunicar. Ao se identificar o público específico interessado nas informações que se dispõe, faz-se necessário quais informações se apresenta.

No contexto do setor aéreo, temos uma grande variedade de público que possa estar interessado nas informações que serão apresentadas, entre eles especialistas, acadêmicos, executivos do setor, profissionais de imprensa e a sociedade com um todo. Na esfera em que trabalharemos, vamos nos atentar especificamente a sociedade, nosso nicho para apresentação dos dados coletados.

Posto isso, a primeira pergunta que cabe é: Em que tipo de informação a sociedade está interessada quando falamos de transporte aéreo de passageiros?

Talvez a primeira resposta que se apresente seja: “Nos preços, é claro.”

Certamente umas das diversas respostas possíveis seria essa. Mas como objetivo está além de simplesmente apresentar dados, informar e explicar a razão de ser deles é também uma premissa da apresentação de dados.

Por isso, este trabalho não se resume a simplesmente apresentar preços ou valores números compilados e resumidos, mas também oferecer a razão se ser de cada um deles, a fim de entender o atual mercado doméstico brasileiro quando ao transporte aéreo de passageiros.

A segunda lição que a autora nos oferece diz respeito a escolha de um visual apropriado. Para o nosso público, o maior dos levantados acima como possíveis interessados, verifica-se uma variedade de níveis de instrução. Para isso é importante oferecer uma visualização que seja o mais limpa possível e apresente de forma separada cada dado compilado. Ao se correlacionar um dado com outro, é possível ainda criar uma forma de história, a fim de apresentar em etapas o que se pretender dizer.

A terceira lição apresentada é: Elimine a saturação. Informações como título, bordas, linhas de grade, legendas, variações de cores diversas não devem estar em primeiro plano na apresentação gráfica. Idealmente, ao se olhar para um gráfico, o interessado na informação deve ver o essencial daquilo que se pretende apresentar, bem como ver a informação relevante para aquela análise, apresentada graficamente.

Na quarta lição, que decorre da anterior, é chamar a atenção do público para o que se pretende focar. O uso de cores, negrito e marcadores. Para isso, conforme apresenta na lição número cinco, é preciso pensar como um designer, de modo que o visual gráfico se torne acessível com texto, usando textos mais simples, que contem a história dos dados, bem como alinhando elementos para melhorar a estética, de modo que o texto centralizado não é uma recomendação, mas comumente utilizado em apresentações de dados.

Por último, a sexta lição nos ensina a escrever os dados como se contasse uma história, e isso é ainda mais importante quando temos vários dados para analisar ou correlacionados.

Existe uma pesquisa encontrada nas pesquisas deste trabalho relacionado ao tema em debate, que tem como objeto o uso do *Storytelling* voltado a dados, porém este trabalho de Farias (2020) tem como norte o uso de dashboards para apresentar e contar a história dos dados analisados.

Há também o trabalho realizado por Formigoni (2021), que relaciona o uso da linguagem de programação Python ao estudo de acidentes aéreos no Brasil, utilizando a extração de dados de voos do site da ANAC.

É importante também mencionar o trabalho de pesquisa realizado por Pereira (2021), que trata da alocação de frota na aviação regional, onde os estudos de coleta de dados são realizados usando linguagem de programação Python. Embora não seja o foco do trabalho o tema do *Storytelling*, este trata da análise de dados do setor da aviação civil, e pode servir de embasamento para outros abordagens sobre o tema

3 MÉTODOS E TÉCNICAS DE PESQUISA

Esta pesquisa foi elaborada a partir da necessidade de desenvolver um método que pudesse extrair e analisar as informações obtidas por meio do site eletrônico da Agência Nacional de Aviação Civil (ANAC), a fim de analisar pontos como Panorama de Mercado, Demanda e Oferta, Tarifas Aéreas, Relatórios Especiais, Atrasos e Cancelamentos, e outros dados relacionados a Passageiros. A ferramenta, realizada através de Python, pode servir de instrumento de mensuração de dados do setor aéreo para tomada de decisão, de forma a subsidiar políticas econômicas, ou mesmo como instrumento para organizações que atuam no setor de turismo de uma maneira geral, compondo uma série de ferramentas para Tomada de Decisão de oferta de produtos e serviços, de acordo com cada localidade analisada, utilizando a ferramenta de linguagem de programação Python.

Esta pesquisa é do tipo exploratória, cuja aérea de estudo é a aviação, utilizando o método de análise de séries temporais, como instrumento de pesquisa será utilizado o *Design Science.*

A coleta dos dados foi obtida através da plataforma da Agência Nacional de Aviação Civil (ANAC) e a análise dos dados coletados foi executada com a ferramenta de linguagem de programação Python, que permite uma maior personalização de leitura de dados, por meio de gráficos e dashboards customizáveis.

Esta pesquisa foi elaborada utilizando o método de *Design Science.* Este método foi concebido por Herbert Simon em 1969. Seu conceito central, descrito em sua obra intitulada *As Ciências do Artificial*, Simon (Dresch, 2019), faz uma diferença entre o natural e o artificial, quando trata de objeto de pesquisa.

De acordo com Simon (Dresch, 2019) o *Design Science* pode ser definido como a ciência do projeto, e tem como foco o estudo de como criar e projetar, a fim de ajudar na solução de problemas reais, não alcançados pelas ciências naturais, que são base das pesquisas em engenharia.

Assim, o caráter prescritivo desta abordagem metodológica pode ajudar a resolver problemas que as ciências naturais e sociais não foram capazes de apresentar solução possível, de tal maneira que pudessem ser executadas.

A importância desta abordagem metodológica para as pesquisas em engenharia é tornar o conhecimento desenvolvido útil para a prática das organizações, e da sociedade, visto que a abordagem tradicional em engenharia, por vezes, não apresenta soluções práticas para

implementação e/ou resolução de problemas cotidianos, por muitas vezes se pautar exclusivamente em ciências naturais.

É importante mencionar os fundamentos do *Design Science* conforme aponta Simon (apud DRESCH, 2019), calcado em uma natureza prática de solução de problemas. Entre os fundamentos do *Design Science* estão: i) natureza da pesquisa; ii) os artefatos produzidos; iii) as soluções satisfatórias geradas; iv) a validade pragmática da solução proposta; e v) as classes de problemas que organizam a trajetória do conhecimento no âmbito da *Design Science*.

Os artefatos elaborados sob a ótica do *Design Science* recebem a classificação de: constructos, modelos, métodos, instanciações (MARCH; SMITH, 1995 apud DRESCH, 2019).

Abaixo é apresentado o quadro 3.1, em que o conceito de cada classificação aparece:

Tabela 1 - Tipos de Artefatos na pesquisa em engenharia

Tipo de artefato	Definição	Exemplo de artefato desenvolvido	Referências dos exemplos
Constructo*	Conceitos utilizados para descrever problemas ou especificar as respectivas soluções	Constructos para descrever os fluxos de informação de um processo de negócio	Rosenkranz e Holten (2011)
Modelo	Conjunto de elementos e relações que representam a estrutura geral da realidade	Modelo de controle integrado da produção e da qualidade para construção civil	Leão, Isatto e Formoso (2016)
Método	Conjunto de passos lógicos necessários para a efetivação de determinada atividade	Método para resolver desafios concernentes à arquitetura organizacional	Nakakawa, Van Bommel e Proper (2011)
Instanciação	Execução do(s) artefato (s) em seu ambiente real, evidenciando a viabilidade e eficácia dos artefatos	Instanciação de um método para desenvolver habilidades de resiliência em eletricistas	Saurin *et al.* (2014)
Design Proposition	Regras tecnológicas ou regras de projeto, consideradas contribuições teóricas da *Design Science*.	*Design proposition* para uma abordagem de inovação aberta para micro e pequenas empresas	Krause e Schutte (2016)

Fonte: Dresch, Metodologia Científica para Engenharia, Cap. 5: Fundamentos do Design Science e da Design Resarch para a engenharia.

A abordagem de *Design Science* e de *Design Science Resarch* requer a elaboração de mapas causais para análises das situações levantadas para as quais se busca uma solução. Este estudo requer a colaboração dos agentes envolvidos e está embasada em criação de artefatos, a fim de criar o que Druckenmiller e Acar (2009) chamaram de instanciação do Comprehensive Situation Mapping (CSM). Isto auxilia na elaboração de mapas estratégicos para elucidar soluções para determinado problema levantado na pesquisa.

Este método está orientado por Hevner (2004, apud Dresch, 2009), e que indica sete critérios que devem ser considerados para a condução de uma pesquisa com Design Science Research, são eles:

i) Comunicação da pesquisa;

ii) Rigor;

iii) Relevância do problema;

iv) Design como artefato;

v) Design como um processo de pesquisa;

vi) Avaliação do design;

vii) Contribuição do design.

Cada um desses critérios foi desenhado para garantir o rigor que o Design Science Research exige para desenvolver o seu método.

No campo da pesquisa em questão, 12 etapas serão desenvolvidas, de acordo com o *Design Science Research.* São elas:

1. Identificação do problema:

Nesta etapa o problema endereçado é apresentado, mostrando-se uma preocupação com o tema e a relevância do seu estudo da análise de demanda por transporte aéreo no Brasil, a fim de identificar quais fatores de correlação poderiam prever um movimento de maior ou menor demanda no setor.

Ao mesmo tempo em que os problemas da questão levantada são analisados, quais sejam, as dificuldades dos profissionais da aérea e organizações para tomada de decisão quanto a apropriação da informação adequada sobre as demandas por transporte aéreo.

2. Conscientização do problema:

Os elementos necessários para tomada de decisão poderão ser abordados aqui, de modo que se saiba a razão de determinado problema existir, que em questão se apresenta como a necessidade de maior precisão na oferta de transporte aéreo, a fim de diminuir o os inconvenientes de cancelamentos em função da reprogramação de rotas por não obter suficiente demanda para determinado trecho.

3. Identificação de artefatos e configuração das classes de problemas.

Os artefatos apresentados podem servir de instrumento para tomada de decisão para organizações do setor aéreo, como agências de turismo, de modo que ao prever demandas, possam oferecer mais serviços online para potenciais consumidores.

Artefatos identificados:

(i) teoria dos grafos;

(ii) diagramação de influência;

(iii) mapas causais;

(iv) dinâmica de sistemas;

(v) CSM (Comprehensive Situation Mapping)

4. Proposição de artefatos para resolver o problema específico;

Neste tópico, possuímos três artefatos, são eles: o método, o modelo, e a instanciação.

Quanto ao método, este será o CSM, que será aplicado de modo que as variáveis de decisão serão escolhidas durante a Análise Exploratória de Dados, na etapa de Elaboração do Trabalho de Conclusão.

4.1 O CSM é selecionado como um método a ser aplicado na instanciação, a ser adaptado e instanciado para solucionar o problema que está sendo endereçado, de forma que o apontamento será apresentado graficamente. Nesta pesquisa, CSM serviu como um método em que se desenvolveu um modelo para aplicar em dados, em uma instanciação.

A escolha do CSM é justificada com base nas vantagens que este método apresenta em relação às demais abordagens, identificando períodos específicos nos quais a informação é relevante para tomada decisão de empresas do setor.

Vantagens do CSM:

(i) combinação de métodos analíticos e dialéticos para a compreensão do problema e a construção dos cenários estratégicos da organização;

(ii) combinação de técnicas analíticas quantitativas e percepções qualitativas dos participantes;

(iii) possibilidade de formalização das divergências existentes entre os participantes no estabelecimento de cenários e formulação de estratégias;

(iv) capacidade de orientação para a convergência das visões dos participantes.

6) Projeto do artefato selecionado

i) São detalhados cada um dos elementos do CSM, a fim de assegurar sua posterior instanciação

ii) Um conjunto de estudos que aplicaram o CSM é consultado, para identificar as suas limitações e orientar o projeto da instanciação

iii) São identificadas e selecionadas técnicas e ferramentas para assegurar uma solução satisfatória da instanciação

7) Desenvolvimento do artefato

i) É apresentado o desenvolvimento do artefato até seu estado funcional.

ii) São apresentadas as ferramentas que suportaram o desenvolvimento

8) Avaliação do artefato

i) Realizada com base nos critérios estabelecidos durante a etapa do projeto

ii) Até o atingimento da solução satisfatória, houve iteração entre as etapas de avaliação e desenvolvimento.

Critérios utilizados:

(i) interface do usuário, analisando sua eficácia, capacidade de gerar aprendizagem e facilidade no manuseio;

(ii) capacidade para estabelecer os dados da baseline para a realização de tarefas comuns;

(iii) compilação das tarefas previstas, gerando melhorias posteriores no artefato

9) Explicitação das aprendizagens

Aprendizagens:

iterações nas etapas de desenvolvimento e avaliação; problemas e soluções do processo de pesquisa.

10) Conclusões

Principais contribuições e resultados obtidos com instanciação do CSM foram apresentados. As limitações do estudo são evidenciadas. São propostas sugestões para pesquisas futuras para refinamento do CSM.

Para desenvolver a execução da ferramenta será utilizado a plataforma Google Colab. Como aponta Santos (2022) O Google Colab, ou Google Colaboratory é uma ferramenta que possibilita que se misture código fonte (em geral em linguagem de programação Python) e texto rico (markdown: linguagem voltada para formatação de textos) com imagens e o resultado do código desenvolvido. A essa mistura denomina-se o que se verá citado à frente como Notebook. Este serviço é gratuito, hospedado pelo Google, e é desenvolvido na nuvem. Tem como objetivo a pesquisa de aprendizado de Máquina e Inteligência Artificial.

3.1 Tipologia e descrição geral dos métodos de pesquisa

O tipo de pesquisa é de Análise Exploratória de Dados, usando a ferramenta de linguagem de programação Python. Para criação de códigos para esta pesquisa será utilizado a plataforma virtual GitHub, portal de hospedagem de códigos para controle de versão e colaboração. O portal permite que pessoas trabalhem colaborativamente no desenvolvimento de códigos para diferentes ferramentas, entre elas a Análise Exploratória de Dados.

3.2 Caracterização da organização, setor e área

O objeto de estudo desta pesquisa é o mercado de transporte aéreo no Brasil. Especificamente, levantaremos os dados coletados e produzidos pela Agência Nacional de Aviação Civil do Brasil.

Em anexo a este trabalho consta e Lei nº 11.182, de 27 de dezembro de 2005, que cria a Agência Nacional de Aviação Civil. Também o Decreto nº 5.731, de 20 de março de 2006, que dispõe sore a instalação, a estrutura organizacional da ANAC e aprova o seu regulamento. O Regimento Interno da ANAC, através da Resolução nº 381, de 14 de junho de 2016. A Resolução nº 581, de 21 de agosto de 2020, que altera o regimento interno da agência. A Instrução Normativa nº 127, de 05 de outubro de 2018, que estabelece regras e diretrizes para a edição do regimento interno, para a organização interna das unidades organizacionais e para os processos de modificação da estrutura organizacional da Agência. Bem como as seguintes portarias, que criam a organização das superintendências da Agência Nacional de Aviação Civil:

- ✓ Portaria nº 1.000/2019 – Organização Interna da Superintendência de Administração e Finanças - SAF
- ✓ Portaria nº 2.228 /2019 - Organização Interna da Superintendência de Planejamento Institucional - SPI
- ✓ Portaria nº 2.293/2019 - Organização Interna da Superintendência de Tecnologia da Informação - STI
- ✓ Portaria nº 2.754/2019 – Organização Interna da Superintendência de Ação Fiscal - SFI
- ✓ Portaria nº 2.928 /2020 - Organização Interna da Superintendência da Superintendência de Pessoal da Aviação Civil - SPL
- ✓ Portaria nº 3.059/2020 - Organização Interna da Superintendência de Regulação Econômica de Aeroportos - SRA
- ✓ Portaria nº 3.881/2020 - Organização Interna da Superintendência de Aeronavegabilidade - SAR
- ✓ Portaria nº 3.901/2020 - Organização Interna da Superintendência de Infraestrutura Aeroportuária - SIA
- ✓ Portaria nº 4.062/2021 - Organização Interna da Superintendência de Gestão de Pessoas - SGP
- ✓ Portaria nº 4.211/2021 - Organização Interna da Superintendência de Acompanhamento de Serviços Aéreos - SAS
- ✓ Portaria nº 4.919/2021 - Organização Interna da Superintendência de Padrões Operacionais – SPO

3.3 População e amostra da pesquisa

A população analisada é a do mercado de transporte aéreo de passageiros no Brasil, e que está regulado pela ANAC.

A ANAC tem a prerrogativa de regular o setor de transporte aéreo no Brasil, editar normas e decretos sobre o processo de regulação das atividades do setor, diretrizes e procedimentos para garantir a segurança do transporte aéreo no país, bem como fiscalizar a infraestrutura aeronáutica e aeroportuária do setor.

Dos dados coletados pela ANAC, uma amostra será selecionada para análise. Entre os dados que podem ser analisados usando a ferramenta proposta estão disponíveis em Dados e Estatísticas, Mercado de Transporte Aéreo, Passageiros.

3.4 Caracterização e descrição dos instrumentos de pesquisa

Esta pesquisa é caracterizada pelo uso do instrumento de raspagem de dados e pesquisa dos dados coletados pela ANAC, uma amostra será selecionada para análise. Entre os dados que podem ser analisados usando a ferramenta proposta estão disponíveis em Dados e Estatísticas, Mercado de Transporte Aéreo, Passageiros.

3.5 Procedimentos de coleta e de análise de dados

Os procedimentos para coleta dos dados se dará por meio do acesso ao site da ANAC, na área de Dados e Estatísticas. Portanto a amostra desta pesquisa será por conveniência.

A coleta ocorrerá durante a segunda etapa desta pesquisa, que se iniciará com a Análise Exploratória de Dados. Os dados estão dispostos em planilhas Excel. Em seguida estes dados serão orientados usando Notebooks desenvolvidos na etapa de programação dos instrumentos.

As variáveis coletadas são de valor quantitativo, que permitem maior manipulação e avaliação estatística.

4 RESULTADOS E DISCUSSÃO

Além das particularidades elencadas acima, que podem ser apresentadas de maneira didática usando a linguagem de programação Python, a forma de apresentar dados não deve levar em consideração apenas o dado em si, mas o público que o visualizará, bem como a história que se pretender contar, no sentido de qual esclarecimento se quer dar a determinado público. Assim, uma mesma informação pode ser apresentada de diferentes maneiras, a depender de qual público irá ouvi-lo.

Nas palavras de Knaflic (2019) “Se é difícil de ler, é difícil fazer”. Esta foi a conclusão a que se chegou uma pesquisa feita por Song e Schwarz na Universidade de Michigan, em 2008, quando foi entregue a dois grupos de alunos uma mesma atividade em que as instruções foram escritas em fonte Arial e a outra metade em fonte cursiva chamada Brushstroke. Foi perguntado aos alunos quanto tempo demorariam para realizar o exercício, e quanto mais difícil fosse a escrita, maior a probabilidade de não realizarem a tarefa, como expôs KNAFLIC (2019).

Por essa razão, apresenta-se três formas de visualização de dados usando a linguagem de programação Python, são estas:

- Visualização com Pandas
- Visualização com *Matplotlib*
- Visualização com *Seaborn*

Pandas é uma biblioteca de dados que oferece ferramentas de análise e estrutura de dados (FIGUEIREDO, 2018 apud FERREIRA, 2021). Ela permite uma abordagem fácil de usar, e é uma das bibliotecas mais completas para o objetivo da análise de dados, tornando-se fundamental para esse fim. É possível encontrar como importar a biblioteca de dados para o projeto de análise de dados no apêndice deste docuemnto, página 37, no Notebook I (para saber o que é um Notebook do Google Colab, volte aqui).

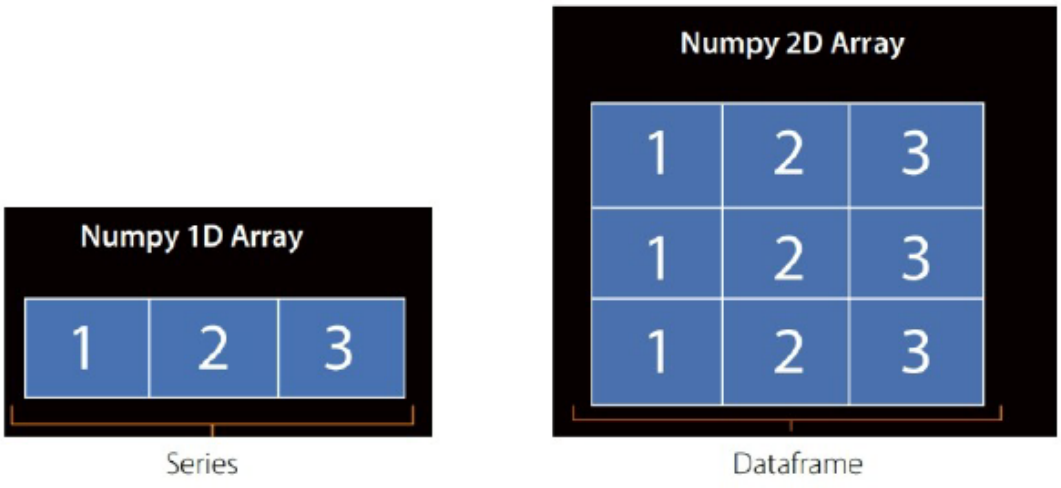
Uma função fundamental para aplicação desta ferramenta foi o pacote *Selenium,* que é um banco de dados que permite que o dispositivo utilizado para aplicação utilize o navegador para buscar os dados de forma automatizada.

A função básica para sua aplicação é *pip install -U Selenium.* Ela poderá ser mais bem avaliada nas páginas 36 e 37 do Notebook I, pois apresenta uma série de passos, desde a abertura das páginas web, até a importação do módulo *webdriver,* que provê as implementações para diversos browsers. Em seguida, mostra-se a manipulação dos diversos elementos de uma página, como botões, listas suspensas etc.

A biblioteca Pandas fornece uma estrutura de dados rápida e flexível, projetadas para facilitar o trabalho com dados relacionais ou rotulados, usando manipulação de código aberto, construída sobre a linguagem de programação Python, segundo Pandas (2022).

Existem duas principais estruturas de dados do Pandas, são as Series, denominada unidimensional e a DataFrame, dita bidimensional, como ilustra a Figura 3.

Figura 3 – Series e DataFrame



Fonte: Adaptada de Preparação e análise de dados (2021).

Igualmente, a Serie e o DataFrame são chamados de Numpy Array. Pandas é construído sobre Numpy, que se trata de uma biblioteca Python, utilizada para fazer cálculos em arrays de várias dimensões. Ela é capaz de fornecer um grande conjunto de funções e operações que colaboram no trabalho de programadores quando se trata de trabalhos majoritariamente numéricos, como aborda Ferreira (2021).

O Pandas permite importar diversos formatos de arquivo, como o *.csv,. xls, json, .html* e outros. Na presente análise foi utilizado o formato .csv.

Como exposto acima, o fato de o documento conter uma quantidade muito grande de dados, tornando a extração bastante demorada, se decidiu por analisar apenas o ano de 2022, até o mês de maio.

O processo para se analisar um, dois ou cinco anos é o mesmo, pois trataremos aqui em como apresentar esta compilação de dados de acordo com o método apresentado por Knaflic (2019).

Segundo Ferreira et al. (2021) a finalidade da análise exploratória de dados (AED) é checar os dados para qualquer explicação estatística. A partir disso, se obtém o entendimento sobre os dados coletados, bem como sobre as relações existentes entre as variáveis manipuladas.

Na abordagem AED, a estrutura para solução de um problema é elaborada a partir da seguinte etapa:

Problema > Dados > Análise > Modelo

Após levantarmos o problema, os dados foram obtidos junto a ANAC, através da extração de dados usando o código disponível na página 41 no Notebook I, e a análise foi

realizada de acordo com os critérios necessários de análise, quais sejam o de identificar *insights*, para identificação de padrões nos dados analisados.

Seaborn é uma biblioteca de visualização de dados da linguagem de programação em Python baseada na biblioteca de plotagem *Matplotlib (pág. 54 – Notebook II)*, como afirma Ferreira (2021). Esta biblioteca disponibiliza diversos gráficos e interfaces interessantes e de fácil compreensão quando se trata de plotagem de dados. Além disso, ela contribui no processo de AED, auxiliando na visualização de dados. A importação da biblioteca pode ser realizada por meio do código na página 54 no Notebook II.

Com o *Seaborn* é possível criar gráficos de barras e histogramas customizáveis, auxiliando no processo de análise de dados. Conforme argumenta Knaflic (2019) os gráficos de barras são mais fáceis de compreender do que os gráficos de pizza, dado que para o leitor, eles apresentam uma lógica mais compreensível para se observar um dado.

Também é altamente recomendado o uso de gráficos de uma única cor, com gradações ou não, isso permite que a linguagem gráfica seja mais limpa para visualização, tornando mais claro o que se pretende informar.

Abaixo é apresentado os resultados dos dois notebooks utilizados para extração e análise dos dados da agência. Como foi dito, com a atual ferramenta, é possível analisar qualquer período, bem como extraí-los do site da ANAC. No arquivo em questão foi analisado no ano de 2022, até o quinto mês do ano, o que já representou uma grande quantidade de dados extraído, se se considerar que se está falando de cada trecho percorrido dentro do setor aéreo doméstico.

É importante mencionar que dois bancos de dados foram utilizados, o da ANAC, extraído usando a ferramenta, bem como o *airportsdata (pág. 44 – Notebook I)*, um extensivo banco de dados de quase todos os aeroportos e pistas de pouco do mundo, com mais de 28 mil entradas.

Esse banco de dados foi fundamental, pois há alguns aeroportos com grande movimento que não estão registrados no banco de dados da ANAC, como o aeroporto de Vitória da Conquista (SBQV) e de Bauru Arealva (SJTC), por exemplo.

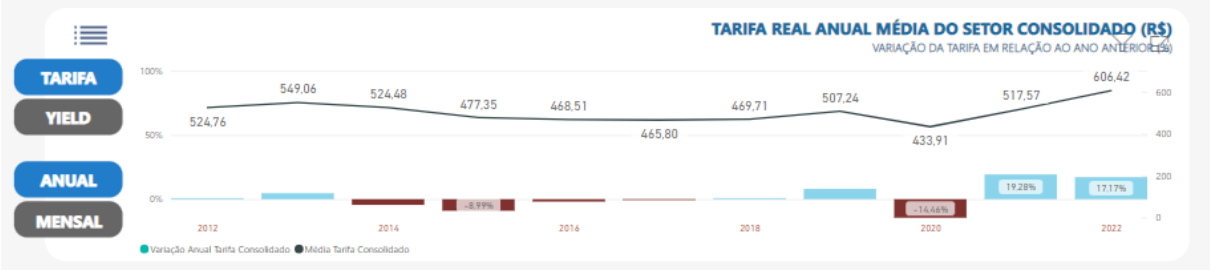
Os resultados gráficos podem ser visualizados a partir da página 52 deste documento, onde foi criado o Notebook II, denominado Análise Exploratória de Dados (AED) em Python.

No Notebook II se verifica diversos estilos gráficos usando as bibliotecas *Pandas*, *Seaborn* e *Matplotlib*, a partir da página 63 no Notebook II.

Desta forma, pretendia-se apresentar como proposta o seguinte exemplo, usando os conceitos aqui abordados sobre *Storytelling* com dados:

Atualmente, se vê a seguinte informação no site da ANAC:

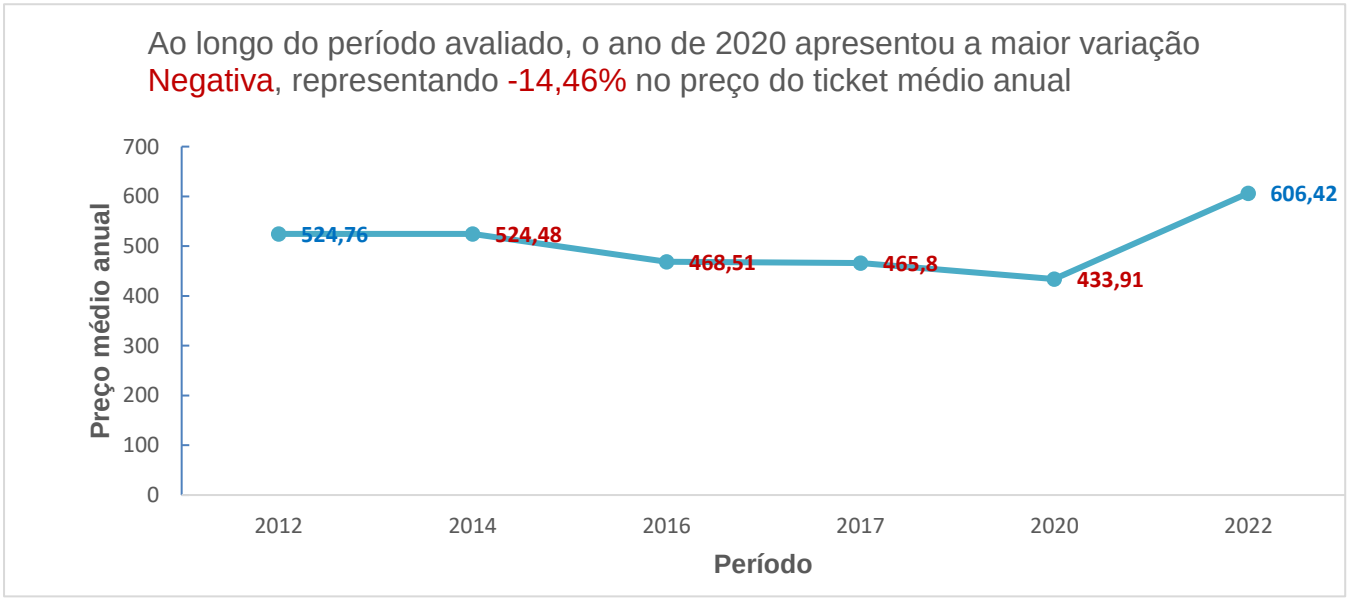
Figura 4 – Tarifa real anual média do setor consolidado (R$)



Aparentemente, observa-se que os dados foram apresentados de forma conjunta, tanto em relação a variação real média do *ticket* médio, quando a sua variação percentual ao longo do período analisado.

De outra forma, poderíamos abordar a mesma informação da seguinte maneira, facilitando a compreensão do leitor, bem como trazendo o conceito de se contar a história dos dados de forma a elucidá-los o melhor possível:

Figura 5 – Como a tarifa do *ticket* real médio anual evoluiu ao longo do tempo



Embora pareça muito mais simples do que o modelo apresentado pela Agência, este gráfico mostra de maneira clara aquilo que se pretende apresentar ao leitor. Este modelo ainda apresenta uma variável interessante. Ele mostra em azul o preço médio anual do *ticket* no primeiro ano analisado, bem como do último, mostrando ao leitor a variação real total do período, trazendo ainda outros *insights* sobre a informação apresentada.

O apresentador ainda poderia frisar a variação positiva do período, apresentando como ponto de partida a retomada da demanda por transporte aéreo de passageiros no ano de 2022, assim como as variáveis relacionadas ao aumento do preço do *ticket* médio.

Ao final deste trabalho apresenta-se uma proposta inicial de como este modelo poderia ser exposto usando a linguagem de programação Python. Também não há impedimentos para o uso de instrumentos auxiliares na formulação da proposta aqui inicialmente apresentadas. Este assunto jamais se encerraria por aqui, ao contrário, é um ensejo a um estudo mais aprofundado sobre esta temática dentro do setor aéreo brasileiro.

Consta no Apêndice desta pesquisa dois Notebooks produzidos na ferramenta Google Colab prontos para serem executados com o fim de extrair, organizar, analisar e plotar os dados coletados, a partir dos dois bancos de dados utilizados, que se complementaram para compor a versão final, graficamente. Essas duas ferramentas podem nortear novas propostas, tornando a gestão da análise dos dados da aviação civil no Brasil mais modelável, sucinta, e de fácil compreensão à sociedade brasileira.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O objetivo deste trabalho em compor um instrumento de extração e leitura da base de dados da Agência Nacional de Aviação Civil foi cumprido, sendo oferecido dois Notebooks que constam no Apêndice deste trabalho. O Notebook II oferece exemplos gráficos claros e simples, que otimizam a leitura da informação mencionada.

A necessidade de se criar uma ferramenta para extrair, analisar e projetar esses dados, a fim de apresentar um cenário que seja legível e compreensível para qualquer um se norteou esta pesquisa, e é apresentado no Notebook I deste documento.

Sendo assim, os três objetivos específicos elencados nesta pesquisa foram alcançados, sendo eles:

1. Criar um modelo de Notebook em Python que extraia os dados do portal ANAC e gere um modelo gráfico de fácil compreensão;
2. Criar um Notebook que analise os dados extraídos;
3. Apresentar graficamente os resultados.

Eles se confirmam na exposição dos modelos de extração apresentados no apêndice.

Pretendeu-se oferecer uma proposta de pesquisa para que a agência se sirva de mais uma possibilidade de instrumento de mensuração da qualidade de apresentação das informações para o setor da aviação civil, ou outros agentes públicos interessados que possam se utilizar desta ferramenta de forma gratuita.

Uma nova abordagem da EDA pode contribuir e enriquecer, trazendo o interesse de outros pesquisadores voltados ao tema ora discutido, de forma a criar conteúdo sobre o assunto,

a fim de inspirar também trabalhos futuros na melhoria da apresentação das informações ora exploradas.

Os objetivos deste trabalho perpassam também por contribuir socialmente, estimulando o acesso à informação de qualidade, já tratada e condicionada, usando princípios da análise de dados estatísticos por meio da linguagem de programação Python.

Os modelos de Notebook desenhados em Python podem ser oferecidos à ANAC, para que esta exploração se enriqueça, e a informação à sociedade se torne altamente disponível, e de fácil compreensão.

Houve algumas limitações consideráveis para esta pesquisa, entre elas a dificuldade de se gerar um relatório completo de cinco anos (2017-2021) devido a imensa quantidade de dados do período. O tempo de extração destas informações era bastante elevado, e a ferramenta do Google Colab se desconecta com frequência, não permitindo a continuação deste trabalho de organização das informações.

Este trabalho se oferece também como uma sugestão para novas pesquisas para o setor da aviação civil, de movo a inspirar outras buscas relacionadas à linguagem de programação utilizada. Bem como uma sugestão para aperfeiçoar o processo de extração de dados em grande quantidade, automatizando todo o processo.

REFERÊNCIAS BIBLIOGRÁFICAS

AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL (ANAC). **Acesso à informação: Institucional**. Brasília, 04/2022. Disponível em: <https://www.gov.br/anac/pt-br/acesso-a-informacao/institucional> Acesso em 24 de abril de 2022.

AIRPORTSDATA. **Extensive database of location and timezone data for nearly every airport and landing strip in the world**. Brasília 06/2022. Disponível em<https://pypi.org/project/airportsdata/>. Acesso em 25 de junho de 2022.

ALURA. **Google Colab: o que é, tutorial de como usar e criar códigos**. Brasília 09/2022. Disponível em: <https://www.alura.com.br/artigos/google-colab-o-que-e-e-como-usar>. Acesso em 29 de setembro e 2022.

ANAC. **Lista de Aeródromos Públicos**. Brasília 06/2022. Disponível em: <https://www.anac.gov.br/acesso-a-informacao/dados-abertos/areas-de-atuacao/aerodromos/lista-de-aerodromos-publicos-v2> Acesso em: 1º de junho de 2022.

ANAC. **Dados Abertos**. Brasília 05/2022. Disponível em: < https://www.anac.gov.br/acesso-a-informacao/dados-abertos/> Acesso em 10 de maio de 2022.

ANAC. Brasília, 04/2022. Tarifas aéreas domésticas. Disponível em: <https://app.powerbi.com/view?r=eyJrIjoiMjJjZjA3YTQtNjYwMi00NjZhLTg5NTUtMzRhODZlN2U0ZTc5IiwidCI6ImI1NzQ4ZjZlLWI0YTQtNGIyYi1hYjJhLWVmOTUyMjM2ODM2NiIsImMiOjR9&pageName=ReportSection7a8d3f66e2d8c1e70619> Acesso em 24 de abril de 2022.

ANAC. Resolução nº 381, de 14 de julho de 2016. Disponível em: <https://pergamum.anac.gov.br/arquivos/ra2016-0381.pdf> Acesso em 12 de abril de 2022.

ANDERSON, D. ... [et al.]. **Estatística Aplicada a Administração e Economia.** 5 ed. São Paulo: Cengage Learning, 2021.

COLAB. **Data Table Display.** Brasília 07/2022. Disponível em: <https://colab.research.google.com/notebooks/data_table.ipynb>. Acesso em 25 de junho de 2022.

DRESCH, A. ... [et al]. **Metodologia Científica para Engenharia**. 1 ed. Rio de Janeiro: Elsevier, 2019.

FARIAS. **Storytelling de Dados: Contando histórias com dashboards**. Palhoça 2020. P. Monografia (Bacharelado em Sistemas de Informação) – Universidade do Sul de Santa Catarina.

FERREIRA, Rafael G C.; MIRANDA, Leandro B. A D.; PINTO, Rafael A.; et al. **Preparação e Análise Exploratória de Dados**. Brasília-DF: Grupo A, 2021. E-book. ISBN 9786556902890.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556902890/>. Acesso em: 20 set. 2022.

FORMIGONI. **Python Na Análise De Dados: Estudo De Caso Com Dados De Acidentes Aéreos No Brasil.** Niterói 2021. P. Monografia (Bacharelado em Engenharia de Produção) - Universidade Federal Fluminense.

GEEKFORGEEKS. **Creating a dataframe from Pandas Series.** Brasília 06/2022. Disponível em: <https://www.geeksforgeeks.org/creating-a-dataframe-from-pandas-series/.>. Acesso em 25 de junho de 2022

GITHUB. **Distance calc sphere**. Brasília 03/2022. Disponível em: <https://github.com/djgroen/flee-release/blob/master/distance_calc.py>. Acesso em 25 de março de 2022.

KNAFLIC, C. N. **Storytelling com dados**. "Um guia sobre visualização de dados para profissionais de negócios." ("Business Intelligence e Análise de Dados para Gestão do Negócio ...") Rio de Janeiro: Altas Books Ed., 2019.

MATPLOTLIB. **Matplotlib: Visualization with Python.** Brasília, 08/2022. Disponível em: < https://matplotlib.org/>. Acesso em 05 de agosto de 2022.

NIST. **Exploratory Data Analysis.** Brasília 05/2022. Disponível em: <https://www.itl.nist.gov/div898/handbook/eda/eda.htm>. Acesso em 08 de maio de 2022.

PANDAS. "**Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.**" ("Pandas Introduction - Fast, Powerful & flexible - Learn Data Insights") Brasília, 09/2022. Disponível em: <https://pandas.pydata.org/> Acesso em 20 de setembro de 2022.

PASSAGEIRO DE PRIMEIRA. **ANAC cria painel interativo para mostrar indicadores de tarifas aéreas**. Brasília, 04/2022. Disponível em: <https://passageirodeprimeira.com/anac-cria-painel-interativo-para-mostrar-indicadores-de-tarifas-aereas/> Acesso em 24 de abril de 2022.

PEREIRA. **Estudo de Metodologia de Otimização para o Problema de Alocação de Frota na Aviação Regional.** Brasília, 2021. P. Monografia (Graduação em Engenharia Aeroespacial) – Universidade de Brasília.

SEABORN. **Seaborn: statistical data visualization**. Brasília, 09/2022. Disponível em: <https://seaborn.pydata.org/#> Acesso em 15 de setembro de 2022.

SELENIUM. **Locating Elements.** Brasília 08/2022. Disponível em: <https://selenium-python.readthedocs.io/locating-elements.html>. Acesso em 25 de agosto de 2022.

SELENIUM. **Python language bindings for Selenium WebDriver.** Brasília, 06/2022. Disponível em: <https://pypi.org/project/selenium/> Acesso em 13 de junho de 2022.

APÊNDICES

Notebook I: ANAC - Dados e Estatísticas com Selenium

ANAC - Dados e Estatísticas com Selenium

Montagem do Google Drive

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:

```
%cd /content/drive/My Drive/Colab Notebooks/
```

```
/content/drive/My Drive/Colab Notebooks
```

In [ ]:

```
!pwd
```

```
/content/drive/My Drive/Colab Notebooks
```

Instalação de Pacotes Python ¶

In [ ]:

```
#from time import sleep
#import time
import os
import sys
from zipfile import ZipFile
```

In [ ]:

```
import pandas as pd
```

Instalação do Selenium

In [ ]:

```
#Automated web browser interaction - https://pypi.org/project/selenium/
#https://www.selenium.dev/selenium/docs/api/javascript/module/selenium-webdriver/chrome.html
!pip install -U selenium
```

In [ ]:

```
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium import webdriver
```

In [ ]:

```
!apt-get update # to update ubuntu to correctly run apt install
!apt install chromium-chromedriver
!cp /usr/lib/chromium-browser/chromedriver /usr/bin
```

In [ ]:

```
sys.path.insert(0,'/usr/lib/chromium-browser/chromedriver')
```

In [ ]:

```
#Definindo as opções - chrome_options
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')

#chrome_options.add_argument("download.default_directory=\\content\\drive\\MyDrive\\Colab Notebooks")

download_folder = "reports"
profile = {"plugins.plugins_list": [{"enabled": False,
                                     "name": "Chrome PDF Viewer"}],
#           "download.default_directory": download_folder,
           "download.extensions_to_open": ""}

prefs = {"profile.default_content_settings.popups": 0,
         "download.default_directory": os.getcwd() + os.path.sep,
         "directory_upgrade": True}

chrome_options.add_experimental_option("prefs", profile)
```

In [ ]:

```
#Comando principal - webdriver chromedriver
wd = webdriver.Chrome('chromedriver',options=chrome_options)
```

Dados de Aeroportos e Distâncias entre Aeroportos

Dados de Aeródromos

- Lista de aeródromos públicos: https://www.anac.gov.br/acesso-a-informacao/dados-abertos/areas-de-atuacao/aerodromos/lista-de-aerodromos-publicos-v2 (https://www.anac.gov.br/acesso-a-informacao/dados-abertos/areas-de-atuacao/aerodromos/lista-de-aerodromos-publicos-v2)

Baixando o arquivo de Aeródromos

In [ ]:

```
wd.get('https://www.anac.gov.br/acesso-a-informacao/dados-abertos/areas-de-atuacao/aerodromos/lista-de-aerodromos-publicos-v2')
```

In [ ]:

```
actions = ActionChains(wd)
```

In [ ]:

```
#https://selenium-python.readthedocs.io/locating-elements.html
opcoesAP = wd.find_elements(By.CLASS_NAME, "summary.url")
for i in opcoesAP:
 print(i.text, opcoesAP.index(i))
```

In [ ]:

```
botaoAP = wd.find_elements(By.CLASS_NAME, "summary.url")[1]
print(botaoAP.text)
```

In [ ]:

```
botaoAP.click()
```

In [ ]:

```
baseAP = wd.find_element(By.PARTIAL_LINK_TEXT, "AerodromosPublicos.csv")
```

In [ ]:

```
link = baseAP.get_attribute('href')
link
```

Out[ ]:

```
'https://sistemas.anac.gov.br/dadosabertos/Aerodromos/Lista%20de%20aer%C3%B3dromos%20p%C3%BAblicos/AerodromosPublicos.csv'
```

In [ ]:

```
!wget --no-check-certificate {link}
```

Acessando o arquivo de Aeródromos

In [ ]:

```
aeroportos = pd.read_csv('AerodromosPublicos.csv',encoding='cp1252', delimiter = ';',sk
iprows=1)
```

In [ ]:

```
aeroportos.shape
```

In [ ]:

```
aeroportos.head()
```

In [ ]:

```
aeroportos.columns
```

In [ ]:

```
aeroportos.isnull().values.any()
```

In [ ]:

```
aeroportos.isnull().sum().sum()
```

In [ ]:

```
aer_coord = aeroportos.drop(columns=['CIAD', 'Nome', 'Município', 'UF',
       'Latitude', 'Longitude','Altitude', 'Operação Diurna',
       'Operação Noturna', 'Designação 1','Comprimento 1', 'Largura 1',
       'Resistência 1', 'Superfície 1','Designação 2', 'Comprimento 2',
       'Largura 2', 'Resistência 2','Superfície 2', 'Situação',
       'Validade do Registro','Portaria de Registro', 'Link Portaria'])
```

In [ ]:

```
aer_coord.shape
```

In [ ]:

```
aer_coord.head()
```

In [ ]:

```
aer_coord.columns = ['oaci','municipio','uf','lat','lon']
```

In [ ]:

```
aer_coord.isnull().sum().sum()
```

In [ ]:

```
aer_coord = aer_coord.dropna()
```

In [ ]:

```
aer_coord.to_csv('aerodromos_anac.csv', index = False, decimal = ',')
```

In [ ]:

```
#Exemplificando como buscar latitude de um determinado código OACI
lat = aer_coord[aer_coord.oaci == 'SBAE']['lat'].tolist()
lat
```

Cálculo da Distância entre Aeródromos

In [ ]:

```
#https://pypi.org/project/airportsdata/
!pip install -U airportsdata
```

In [ ]:

```
import airportsdata
```

In [ ]:

```
#Buscando por código ICAO
airports = airportsdata.load()  # key is ICAO code (default)
print(airports['SBQV'])
```

In [ ]:

```
#Buscando por código IATA
airports = airportsdata.load('IATA')  # key is IATA code, not the default
print(airports['VDC'])
```

```
#Função para calcular distância de grande círculo entre aeroportos
#https://github.com/djgroen/flee-release/blob/master/distance_calc.py
#https://www.johndcook.com/blog/python_longitude_latitude/

import math

def distance_on_unit_sphere(lat1, long1, lat2, long2):

    # Convert latitude and longitude to
    # spherical coordinates in radians.
    degrees_to_radians = math.pi/180.0

    # phi = 90 - latitude
    phi1 = (90.0 - lat1)*degrees_to_radians
    phi2 = (90.0 - lat2)*degrees_to_radians

    # theta = longitude
    theta1 = long1*degrees_to_radians
    theta2 = long2*degrees_to_radians

    # Compute spherical distance from spherical coordinates.

    # For two locations in spherical coordinates
    # (1, theta, phi) and (1, theta, phi)
    # cosine( arc length ) =
    #    sin phi sin phi' cos(theta-theta') + cos phi cos phi'
    # distance = rho * arc length

    cos = (math.sin(phi1)*math.sin(phi2)*math.cos(theta1 - theta2) +
           math.cos(phi1)*math.cos(phi2))
    arc = math.acos( cos )

    # Multiply arc by the radius of the earth
    # in your favorite set of units to get length.
    #arc = arc * 3443.92 #in nautical miles (NM)
    arc = arc * 6378.14 #in KM (1.852 * NM)

    return arc
```

Dados do Mercado de Transporte Aéreo

https://www.gov.br/anac/pt-br/assuntos/dados-e-estatisticas/mercado-do-transporte-aereo (https://www.gov.br/anac/pt-br/assuntos/dados-e-estatisticas/mercado-do-transporte-aereo)

- microdados de tarifas aéreas em https://sistemas.anac.gov.br/sas/downloads/view/frmDownload.aspx (https://sistemas.anac.gov.br/sas/downloads/view/frmDownload.aspx);

ATENÇÃO: os arquivos baixados são .CSV, mas no formato Microsoft Excel comma separated values, portanto, não tem o encoding utf-8. Além disso, o Excel em português salva o CSV separado com ponto e vírgula (';').

Microdados de Tarifas Aéreas

Definindo a URL para acessar com o webdriver

In [ ]:

```
wd.get('https://sistemas.anac.gov.br/sas/downloads/view/frmDownload.aspx')
```

In [ ]:

```
actions = ActionChains(wd)
```

Buscar webpage por ID

Encontrar elementos na página - https://selenium-python.readthedocs.io/locating-elements.html (https://selenium-python.readthedocs.io/locating-elements.html)

In [ ]:

```
#Tema
button1 = wd.find_element(By.ID, "MainContent_listTema")
button1.click()
print(button1.text)
```

In [ ]:

```
#Seleciona: Tarifas - Transporte Aéreo Passageiros Domésticos
element1 = Select(wd.find_element(By.ID, "MainContent_listTema"))
element1.select_by_value(value='14')
```

In [ ]:

```
#Ano
button2 = wd.find_element(By.ID, "MainContent_listAno")
button2.click()
#print(button2.text)
```

In [ ]:

```
#Seleciona Ano: 2022
element2 = Select(wd.find_element(By.ID, "MainContent_listAno"))
```

In [ ]:

```
#Escolhendo manualmente um ano ("value")
#Para evitar eventual proteção do site contra robos
element2.select_by_value(value='2022')
```

In [ ]:

```
#Buscar Arquivos
button3 = wd.find_element(By.ID, "MainContent_btnListaArquivos")
button3.click()
```

In [ ]:

```
#Marcar Todos
button4 = wd.find_element(By.ID, "MainContent_btnMarcar")
button4.click()
```

In [ ]:

```
#Baixar Marcados
button4 = wd.find_element(By.ID, "MainContent_btnBaixar")
button4.click()
```

Identificando o arquivo de tarifas baixado para leitura

In [ ]:

```
filename = [f for f in os.listdir() if "anac" in f]
filename
```

In [ ]:

```
#Escolher qual arquivo
filename[5]
```

In [ ]:

```
filelist = ZipFile(filename[5]).namelist()
for i in filelist:
  print(i, filelist.index(i))
```

In [ ]:

```
dados_ano = [[]]
#for f in range(len(filename)):
for f in range(4,5):
    filelist = ZipFile(filename[f]).namelist()
    for i in filelist:
        dados_ano.append(pd.read_csv(ZipFile(filename[f]).open(i), encoding='cp1252',
                         delimiter = ';',decimal=','))
```

In [ ]:

```
dados_ano.pop(0)
```

In [ ]:

```
dados_ano
```

In [ ]:

```
len(dados_ano)
```

In [ ]:

```
dados_ano[0].columns
```

```
for i in range(len(dados_ano)):
    if len(dados_ano[i].columns) == 8:
        print(i)
        dados_ano[i] = dados_ano[i].drop(columns = ['Unnamed: 0'])
```

In [ ]:

```
for i in dados_ano:
  i.columns = ['Ano','Mes','Empresa','Origem','Destino','Tarifa','Assentos']
```

In [ ]:

```
# Concatenando as listas de dados_ano dos arquivos baixados
dados_2022 = pd.concat(dados_ano, axis=0, ignore_index=False)
```

In [ ]:

```
dados_2022.shape
```

In [ ]:

```
dados_2022.head()
```

In [ ]:

```
dados_2022.info()
```

In [ ]:

```
#Aeroporto de Bauru/Arealva na base somente 'SBAE'
dados_2022.Origem = dados_2022.Origem.replace('SJTC','SBAE')
dados_2022.Destino = dados_2022.Destino.replace('SJTC','SBAE')
```

In [ ]:

```
dados_2022.Mes.unique()
```

In [ ]:

```
dados_2022.Mes.value_counts()
```

In [ ]:

```
dados_2022.to_csv('dados_tarifas_2022.csv', index=False)
```

Criando arquivo com aeroportos de interesse

In [ ]:

```
#Se for preciso ler novamente o arquivo de dados
dados = pd.read_csv('dados_tarifas_2022.csv')
```

```
orig = list(dados.Origem.unique())
dest = list(dados.Destino.unique())
orig.sort() == dest.sort()
```

Out[ ]:

```
True
```

In [ ]:

```
lista_aer = pd.concat([dados.Origem, dados.Destino], axis=0)
```

In [ ]:

```
icao_aer = list(lista_aer.unique())
```

In [ ]:

```
len(icao_aer)
```

Out[ ]:

```
160
```

Verificação de acesso a airports - base da ANAC e Python airportsdata

In [ ]:

```
!pip install -U airportsdata
```

In [ ]:

```
import airportsdata
```

In [ ]:

```
airports = airportsdata.load()
print(airports['SBBV'])
```

In [ ]:

```
for k,v in airports.items():
  if k == 'SBBV':
   print(k, v)
```

```
SBBV OrderedDict([('icao', 'SBBV'), ('iata', 'BVB'), ('name', 'Atlas Brasi
l Cantanhede Airport'), ('city', 'Boa Vista'), ('subd', 'Roraima'), ('coun
try', 'BR'), ('elevation', 276.0), ('lat', 2.8413889408), ('lon', -60.6922
225952), ('tz', 'America/Boa_Vista')])
```

In [ ]:

```
airports['SBBV']['icao']
```

```
aer_coord = pd.read_csv('aerodromos_anac.csv', decimal = ',')
```

In [ ]:

```
#Verifica quantos aeródromos não constam da base da ANAC
contador = 0
aer_nao_anac = []
for i in icao_aer:
    j = aer_coord[aer_coord['oaci'] == i]
    if len(j) == 0:
        contador += 1
        aer_nao_anac.append(i)
print(contador)
aer_nao_anac
```

In [ ]:

```
#Cria uma lista de aeroportos que não estão
#nem na Base da ANAC nem no pacote airportsdata
airports = airportsdata.load()
aer_not_ok = []
for i in aer_nao_anac:
  try:
    airports[i]['icao']
    print(i, 'OK')
  except:
    aer_not_ok.append(i)
print(aer_not_ok, 'Not OK')
```

In [ ]:

```
#Revisa a lista icao_aer excluindo aeroportos Not OK
# SBFE: Feira de Santana/BA
# SBRG: Rio Grande/RS
# SBGS: Ponta Grossa/PR
# SBUY: Porto Urucu - Coari/AM
icao_aer = [x for x in icao_aer if x not in aer_not_ok]
len(icao_aer)
```

Construindo a base de dados de aeródromos de interesse

```
airports = airportsdata.load()
iata_aer = []
munic_aer = []
uf_aer = []
lat_aer = []
lon_aer = []
for i in icao_aer:
    j = aer_coord[aer_coord['oaci'] == i]
    if len(j) != 0:
        munic_aer.append(j['municipio'].values[0])
        uf_aer.append(j['uf'].values[0])
        lat_aer.append(j['lat'].values[0])
        lon_aer.append(j['lon'].values[0])
    else:
        munic_aer.append(airports[i]['city'])
        uf_aer.append(airports[i]['subd'])
        lat_aer.append(airports[i]['lat'])
        lon_aer.append(airports[i]['lon'])
    try:
        iata_aer.append(airports[i]['iata'])
    except:
        iata_aer.append('')
```

In [ ]:

```
#https://www.geeksforgeeks.org/creating-a-dataframe-from-pandas-series/
icao_series = pd.Series(icao_aer)
iata_series = pd.Series(iata_aer)
munic_series = pd.Series(munic_aer)
uf_series = pd.Series(uf_aer)
lat_series = pd.Series(lat_aer)
lon_series = pd.Series(lon_aer)
frame = {'icao': icao_series,
         'iata': iata_series,
         'municipio': munic_series,
         'uf': uf_series,
         'lat': lat_series,
         'lon': lon_series}
aerodromo = pd.DataFrame(frame)
```

In [ ]:

```
aerodromo.shape
```

In [ ]:

```
aerodromo.head()
```

In [ ]:

```
aerodromo.isnull().values.any()
```

In [ ]:

```
aerodromo[aerodromo.icao == 'SBBV']
```

```
aerodromo.to_csv('aerodromos.csv', index = False, decimal = ',')
```

Criando arquivo com distância entre aeroportos

In [ ]:

```
aerodromo = pd.read_csv('aerodromos.csv', decimal = ',')
```

In [ ]:

```
aerodromo.head()
```

In [ ]:

```
#Lista igual a icao_aer
aerod = aerodromo.icao.tolist()
```

In [ ]:

```
orig_aer = []
dest_aer = []
dist_aer = []
for i in aerod:
  for j in aerod:
    if i != j:
        o = aerodromo[aerodromo.icao == i]
        d = aerodromo[aerodromo.icao == j]
        lat1 = o['lat'].values[0]
        lon1 = o['lon'].values[0]
        lat2 = d['lat'].values[0]
        lon2 = d['lon'].values[0]
        distancia = distance_on_unit_sphere(lat1, lon1, lat2, lon2)
    else:
        distancia = 0.0
    orig_aer.append(i)
    dest_aer.append(j)
    dist_aer.append(distancia)
```

In [ ]:

```
orig_series = pd.Series(orig_aer)
dest_series = pd.Series(dest_aer)
dist_series = pd.Series(dist_aer)
frame2 = {'orig': orig_series,
          'dest': dest_series,
          'dist': dist_series}
distancias = pd.DataFrame(frame2)
```

In [ ]:

```
distancias.head()
```

```
distancias.to_csv('distancias_2022.csv', index = False, decimal = ',')
```

Incluindo coluna com distância entre aeroportos

Primeira rodada para incluir a distância

In [ ]:

```
#Na primeira rodada
dados = pd.read_csv('dados_tarifas_2022.csv')
```

In [ ]:

```
distancias = pd.read_csv('distancias.csv', decimal = ',')
```

In [ ]:

```
#Inicializa uma nova coluna em dados na primeira rodada
dados['Distancia'] = pd.Series(dtype='float')
```

In [ ]:

```
#Aeroportos Not OK - Inexistentes na base ANAC e em airportsdata
# SBFE: Feira de Santana/BA
# SBRG: Rio Grande/RS
# SBGS: Ponta Grossa/PR
# SBUY: Porto Urucu - Coari/AM
aer_not_ok = ['SBFE', 'SBRG', 'SBGS', 'SBUY']
```

In [ ]:

```
#Exclui aeroportos Not OK na primeira rodada
dados = dados[(~dados.Origem.isin(aer_not_ok)) & (~dados.Destino.isin(aer_not_ok))]
```

Rodadas Posteriores

In [ ]:

```
#Nas rodadas seguintes
dados = pd.read_csv('tarifas_2022.csv')
```

In [ ]:

```
distancias = pd.read_csv('distancias.csv', decimal = ',')
```

In [ ]:

```
dados.head()
```

In [ ]:

```
dados.shape
```

```
dados.info()
```

In [ ]:

```
dados.isnull().values.any()
```

In [ ]:

```
dados.isnull().values.sum()
```

In [ ]:

```
pd.isna(dados.iloc[4333493,7])
```

In [ ]:

```
distancias.head()
```

In [ ]:

```
distancias.shape
```

In [ ]:

```
#Reduz a base de dados para as Origem-Destino repetitivas
dados_OD = dados.groupby(['Origem','Destino']).size().reset_index().rename(columns={0:
'count'})
dados_OD
```

In [ ]:

```
#TESTE: não considerar no código
dados_OD.Origem.name
```

In [ ]:

```
dados_OD.index.size
```

In [ ]:

```
nome = dados_OD.index[9427]
print(nome, dados_OD.loc[nome][0], dados_OD.loc[nome][1], dados_OD.loc[nome][2])
```

In [ ]:

```
#Rodar apenas parte dos índices
for i in range(0,9428):
  nome = dados_OD.index[i]
  origem = dados_OD.loc[nome][0]
  destino = dados_OD.loc[nome][1]
  indices = dados[(dados.Origem == origem) & (dados.Destino == destino)].index
  if pd.isna(dados[dados.index == indices[0]].iloc[0,7]):
    d = distancias[(distancias.orig == origem) & (distancias.dest == destino)]
    distancia = d.dist.values[0]
    dados.loc[indices, 'Distancia'] = distancia
```

```
#Rodar todos os índices
#Inclui todos os índices de dados.groupby(['Origem','Destino'])
#Alterando a célula anterior, que inclui fatias definidas de índices
for nome in dados_OD.index:
  origem = dados_OD.loc[nome][0]
  destino = dados_OD.loc[nome][1]
  indices = dados[(dados.Origem == origem) & (dados.Destino == destino)].index
  if pd.isna(dados[dados.index == indices[0]].iloc[0,7]):
    d = distancias[(distancias.orig == origem) & (distancias.dest == destino)]
    distancia = d.dist.values[0]
    dados.loc[indices, 'Distancia'] = distancia
```

TESTES - Tentativas desatualizadas - Mais lentas

In [ ]:

```
# orig = dados.Origem.unique()
# dest = dados.Destino.unique()
# len(dest)
```

In [ ]:

```
# for i in orig:
#   for j in dest:
#     indices = dados[(dados.Origem == i) & (dados.Destino == j)].index
#     if len(indices) != 0:
#       if pd.isna(dados[dados.index == indices[0]].iloc[0,7]):
#         d = distancias[(distancias.orig == i) & (distancias.dest == j)]
#         distancia = d.dist.values[0]
#         for k in indices:
#           dados['Distancia'][k] = distancia
```

In [ ]:

```
#Tentativa destualizada - Demora excessiva
# for i in orig:
#   for j in dest:
#     indices = dados[(dados.Origem == i) & (dados.Destino == j)].index
#     if len(indices) != 0:
#       if (i in aer_not_ok) or (j in aer_not_ok):
#         print(i, j, 'Um dos aeroportos em Not OK: ', indices)
#         dados = dados.drop(index = indices)
#       else:
#         try:
#           d = distancias[(distancias.orig == i) & (distancias.dest == j)]
#           distancia = d.dist.values[0]
#           for k in indices:
#             dados['Distancia'][k] = distancia
#         except:
#           print(i, j, 'erro nos indices: ', indices)
```

In [ ]:

```
#Solução antiga - muito lenta - SUBSTITUIDA pela anterior
#for i in range(dados.Ano.size):
#    if pd.isna(dados.iloc[i,7]):
#        origem = dados.Origem[i]
#        destino = dados.Destino[i]
#        j = distancias[(distancias.orig == origem) & (distancias.dest == destino)]
#        distancia = j.dist.values[0]
#        dados['Distancia'][i] = distancia
```

Gravando no Drive

In [ ]:

```
dados.to_csv('tarifas_2022.csv', index = False)
```

Incluindo uma coluna com o yield calculado

In [ ]:

```
dados = pd.read_csv('tarifas_2022.csv', decimal = ',')
```

In [ ]:

```
#Inicializa uma nova coluna em dados
dados['Yield'] = pd.Series(dtype='float')
```

In [ ]:

```
dados['Yield'] = dados['Tarifa'] / dados['Distancia']
```

In [ ]:

```
dados.head()
```

In [ ]:

```
dados.to_csv('tarifas_2022.csv', index = False, decimal = ',')
```

Replicando as linhas com número de assentos maior que um

In [ ]:

```
dados = pd.read_csv('tarifas_2022.csv', decimal = ',')
```

In [ ]:

```
#INCOMPLETO
for i in range(len(dados.Tarifa)):
  if dados.Assentos[i] >= 1:
    for j in range(dados.Assentos[i]):
```

Notebook II: Análise Exploratória de Dados (AED) em Python

Análise Exploratória de Dados (AED) - Exploratory Data Analysis (EDA) em Python

Conceitos

Referência: FERREIRA, R.G.C.; MIRANDA, L.B.A.D.; PINTO, R.A.; AL., E. Preparação e Análise Exploratória de Dados. Grupo A, 2021.

Disponível em https://integrada.minhabiblioteca.com.br/books/9786556902890 (https://integrada.minhabiblioteca.com.br/books/9786556902890).

O que é a Análise Exploratória de Dados (AED) - Exploratory Data Analysis (EDA)? ¶

- https://www.itl.nist.gov/div898/handbook/eda/eda.htm (https://www.itl.nist.gov/div898/handbook/eda/eda.htm) (NIST - National Institute of Standards and Technology)
- https://en.wikipedia.org/wiki/Exploratory_data_analysis (https://en.wikipedia.org/wiki/Exploratory_data_analysis) (Wikipedia)

Segundo o livro de referência (FERREIRA et al, 2021):

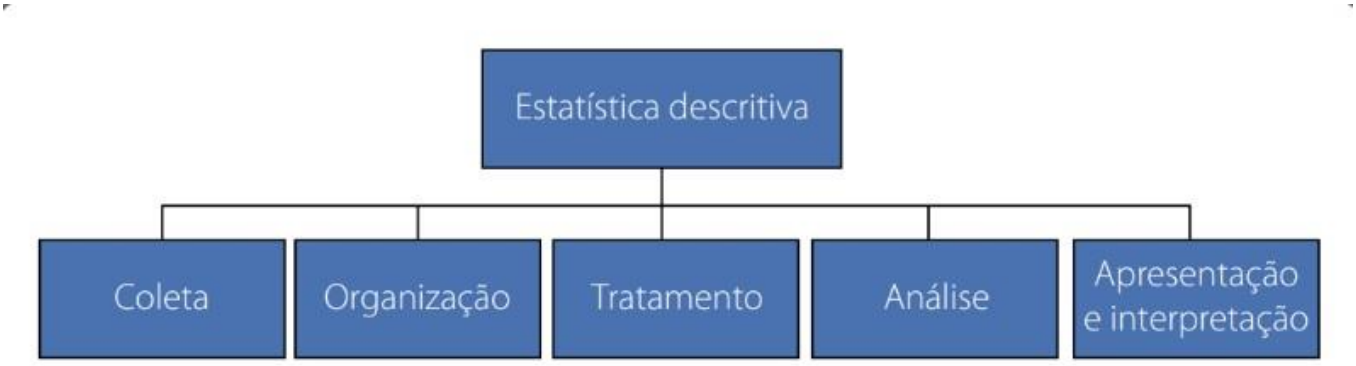
"A finalidade da **análise exploratória de dados (AED)** é verificar os dados para qualquer aplicação estatística. Com isso, obtemos entendimento sobre os dados coletados e, principalmente, sobre as relações existentes entre as variá-veis analisadas. O Quadro 1 apresenta as estratégias utilizadas em diferentes abordagens estatísticas, incluindo a AED".

Quadro 1. Abordagens estatísticas e estratégias utilizadas

Abordagem	Estratégia
Estatística clássica	Problema → Dados → Modelo → Análise
Estatística bayesiana	Problema → Dados → Modelo Priori → Análise
AED	Problema → Dados → Análise → Modelo

Etapas de uma Análise Exploratória de Dados (AED)

"O trabalho da **AED** é geralmente dividido em várias etapas: **coleta**, **organização**, **tratamento**, **análise**, **apresentação** e **interpretação dos dados**. Ressalta-se neste ponto que a *estatística descritiva* está, portanto, fortemente relacionada com o processo de AED, uma vez que as etapas de organização, tratamento, análise e apresentação de dados utilizam técnicas descritivas".



Importância da Análise Exploratória de Dados (AED)

"A **AED** é uma etapa importante para um projeto de *analytics*, pois quebra a ideia de que *data science* é apenas a execução de algoritmos e que deve envolver apenas conceitos de aprendizado de máquina ou técnicas complexas para agregar valor. Toda fase de um projeto de data science pede uma análise exploratória, a qual permite entender o dado, conhecer as suas relações e extrair diversos insights. Uma AED bem feita possibilita encontrar tendências e extrair valor nos dados, incluindo o conhecimento".

Uma base de dados, quando submetida a uma análise, pode conter diferentes problemas:

- **dados ausentes (missing values)**
- **valores discrepantes (outliers)**
- **valores truncados**
- **dados corrompidos**
- **dados incompletos**"
- **pre-processamento (reescalonação, normalização, padronização)**

Pacotes Pyhton utilizados

Pacote Python *pandas*

pacote *pandas* (https://pandas.pydata.org/ (https://pandas.pydata.org/))

Pacote Python numpy

pacote *numpy* (https://numpy.org/ (https://numpy.org/))

Pacote Python matplotlib

pacote *matplotlib* (https://matplotlib.org/ (https://matplotlib.org/))

Pacote Python seaborn

pacote *seaborn* (https://seaborn.pydata.org/ (https://seaborn.pydata.org/))

Importando os pacotes Python

In [ ]:

```
import pandas as pd

import pandas_profiling

import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
sns.set(color_codes=True)
```

Base de Dados - Datasets

Habilitando a opção de Tabelas Dinâmicas no Google Colab

Veja mais detalhes em https://colab.research.google.com/notebooks/data_table.ipynb (https://colab.research.google.com/notebooks/data_table.ipynb)

In [ ]:

```
from google.colab import data_table
data_table.enable_dataframe_formatter()
```

Lendo os Dados no Google Drive

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, c
all drive.mount("/content/drive", force_remount=True).
```

In [ ]:

```
%cd /content/drive/My Drive/Colab Notebooks/
```

```
/content/drive/My Drive/Colab Notebooks
```

```
dados = pd.read_csv('tarifas_2022.csv')
```

In [ ]:

```
aeroportos = pd.read_csv('aerodromos.csv')
```

Utilizando pandas

Conteúdo da base de dados

In [ ]:

```
#permite visualizar o arquivo .csv
#cuidado pois pode demorar para arquivos grandes
!cat tarifas_2022.csv
```

In [ ]:

```
dados.head()
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
0	2022	1	AZU	SBAC	SBAR	552.9	1	718.836149
1	2022	1	AZU	SBAC	SBBR	614.9	1	1675.186803
2	2022	1	AZU	SBAC	SBCA	550.9	1	2826.784854
3	2022	1	AZU	SBAC	SBCF	887.9	1	1804.621127
4	2022	1	AZU	SBAC	SBFL	438.9	3	2812.788957

Dimensão da base de dados

In [ ]:

```
dados.shape
```

Out[ ]:

```
(1871123, 8)
```

Verificando dados faltantes (*missing values*)

In [ ]:

```
dados.isnull().values.any()
```

Out[ ]:

```
False
```

```
dados.isnull().sum()
```

Out[ ]:

```
Ano          0
Mes          0
Empresa      0
Origem       0
Destino      0
Tarifa       0
Assentos     0
Distancia    0
dtype: int64
```

Tratando dados faltantes (missing values)

- fill: preenche com zeros
- drop: exclui linhas com NaN

In [ ]:

```
df = dados.fillna(0)
```

In [ ]:

```
df = dados.dropna()
```

Verificando os tipos das variáveis

In [ ]:

```
dados.dtypes
```

Out[ ]:

```
Ano            int64
Mes            int64
Empresa       object
Origem        object
Destino       object
Tarifa       float64
Assentos       int64
Distancia    float64
dtype: object
```

```
dados.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1871123 entries, 0 to 1871122
Data columns (total 8 columns):
 #   Column     Dtype
---  ------     -----
 0   Ano        int64
 1   Mes        int64
 2   Empresa    object
 3   Origem     object
 4   Destino    object
 5   Tarifa     float64
 6   Assentos   int64
 7   Distancia  float64
dtypes: float64(2), int64(3), object(3)
memory usage: 114.2+ MB
```

Recuperando dados por condição

In [ ]:

```
#Qual a maior tarifa praticada
dados[dados.Tarifa == dados.Tarifa.max()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
1805223	2022	5	TAM	SBBR	SBSP	5998.0	1	873.332384

In [ ]:

```
#Quais tarifas ficaram acima de 5000
dados[dados.Tarifa >= 5000]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
1185436	2022	4	AZU	SBRJ	SBMK	5149.90	1	694.054319
1231825	2022	4	AZU	SBRF	SBAR	5097.68	1	396.235811
1470722	2022	5	AZU	SBGR	SSOU	5579.90	2	712.636752
1519355	2022	5	AZU	SBCF	SSOU	5187.90	2	376.064938
1568408	2022	5	AZU	SBKP	SSOU	5844.90	1	670.829715
1625685	2022	5	AZU	SSOU	SBCF	5187.90	2	376.064938
1627127	2022	5	AZU	SSOU	SBCF	5965.90	1	376.064938
1627131	2022	5	AZU	SSOU	SBKP	5844.90	3	670.829715
1805223	2022	5	TAM	SBBR	SBSP	5998.00	1	873.332384
1810926	2022	5	TAM	SBCY	SBRB	5541.90	1	1431.171760

```
#Qual a menor tarifa praticada
dados[dados.Tarifa == dados.Tarifa.min()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
118672	2022	1	GLO	SBAR	SBGL	21.0	6	1470.922204
118831	2022	1	GLO	SBAR	SBGR	21.0	12	1707.236278
119406	2022	1	GLO	SBBE	SBBR	21.0	6	1613.790004
119853	2022	1	GLO	SBBE	SBEG	21.0	1	1300.283813
120114	2022	1	GLO	SBBE	SBFZ	21.0	16	1137.629914
...	...	...	...	...	...	...	...	...
188685	2022	1	GLO	SBSV	SBRJ	21.0	22	1225.355543
188895	2022	1	GLO	SBSV	SBSG	21.0	2	858.476239
189126	2022	1	GLO	SBSV	SBSP	21.0	40	1482.018047
189424	2022	1	GLO	SBSV	SBVT	21.0	1	844.495366
190461	2022	1	GLO	SBUL	SBGR	21.0	3	538.358312

171 rows × 8 columns

In [ ]:

```
#Quais tarifas ficaram abaixo de 50
dados[dados.Tarifa <= 50]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
118672	2022	1	GLO	SBAR	SBGL	21.00	6	1470.922204
118831	2022	1	GLO	SBAR	SBGR	21.00	12	1707.236278
119406	2022	1	GLO	SBBE	SBBR	21.00	6	1613.790004
119853	2022	1	GLO	SBBE	SBEG	21.00	1	1300.283813
120114	2022	1	GLO	SBBE	SBFZ	21.00	16	1137.629914
...	...	...	...	...	...	...	...	...
1869260	2022	5	TAM	SBGR	SBFZ	43.47	9	2349.283585
1869261	2022	5	TAM	SBRJ	SBSP	48.00	2	366.077306
1870200	2022	5	TAM	SBGR	SBCG	43.47	4	908.250927
1870670	2022	5	TAM	SBGR	SBAR	43.47	1	1707.236278
1870905	2022	5	TAM	SBGL	SBGR	43.47	1	337.169293

1165 rows × 8 columns

```
#Qual a maior distância da base
dados[dados.Distancia == dados.Distancia.max()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
135510	2022	1	GLO	SBCZ	SBJP	1413.90	1	4174.180811
135511	2022	1	GLO	SBCZ	SBJP	1634.90	1	4174.180811
135512	2022	1	GLO	SBCZ	SBJP	1898.90	1	4174.180811
135513	2022	1	GLO	SBCZ	SBJP	2098.90	1	4174.180811
519576	2022	2	GLO	SBCZ	SBJP	1000.90	2	4174.180811
928110	2022	3	GLO	SBCZ	SBJP	2345.90	2	4174.180811
928111	2022	3	GLO	SBCZ	SBJP	2947.90	1	4174.180811
947378	2022	3	GLO	SBJP	SBCZ	1223.90	2	4174.180811
1314164	2022	4	GLO	SBJP	SBCZ	1386.26	1	4174.180811
1356579	2022	4	GLO	SBCZ	SBJP	1918.90	1	4174.180811
1356826	2022	4	GLO	SBJP	SBCZ	1888.90	1	4174.180811
1360662	2022	4	GLO	SBCZ	SBJP	1513.77	1	4174.180811
1663707	2022	5	GLO	SBJP	SBCZ	995.17	2	4174.180811
1663708	2022	5	GLO	SBJP	SBCZ	949.00	1	4174.180811
1677228	2022	5	GLO	SBCZ	SBJP	949.00	1	4174.180811
1711188	2022	5	GLO	SBJP	SBCZ	2128.90	1	4174.180811
1711189	2022	5	GLO	SBJP	SBCZ	750.67	3	4174.180811
1723973	2022	5	GLO	SBCZ	SBJP	995.17	2	4174.180811

In [ ]:

```
#Qual a menor distância da base
dados[dados.Distancia == dados.Distancia.min()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
64461	2022	1	AZU	SBMD	SNYA	60.9	1	65.671788
808994	2022	3	AZU	SBMD	SNYA	272.9	1	65.671788
808995	2022	3	AZU	SBMD	SNYA	60.9	1	65.671788
1591492	2022	5	AZU	SBMD	SNYA	286.9	1	65.671788

```
#Qual a maior número de assentos
dados[dados.Assentos == dados.Assentos.max()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
1692563	2022	5	GLO	SBSP	SBSV	171.9	5465	1482.018047

In [ ]:

```
#Vendas de mais de 1000 assentos
dados[dados.Assentos >= 1000]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
8230	2022	1	AZU	SBBR	SBRF	252.9	1189	1655.869453
38766	2022	1	AZU	SBFZ	SBRF	139.9	1055	627.835645
75208	2022	1	AZU	SBPA	SBRJ	164.9	1259	1121.366501
83290	2022	1	AZU	SBRF	SBBR	252.9	1183	1655.869453
84449	2022	1	AZU	SBRF	SBFZ	139.9	1168	627.835645
...	...	...	...	...	...	...	...	...
1842062	2022	5	TAM	SBCF	SBSP	177.9	1128	524.939419
1842065	2022	5	TAM	SBSP	SBCF	177.9	1233	524.939419
1842463	2022	5	TAM	SBGR	SBNF	210.9	1204	441.620932
1857067	2022	5	TAM	SBSP	SBVT	257.9	1438	756.932106
1864145	2022	5	TAM	SBSP	SBVT	279.9	1039	756.932106

108 rows × 8 columns

In [ ]:

```
#Obtendo uma base de dados da empresa LATAM
TAM = dados.groupby('Empresa').get_group('TAM')
```

In [ ]:

```
#Obtendo a tarifa máxima desta empresa
TAM[TAM.Tarifa == TAM.Tarifa.max()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
1805223	2022	5	TAM	SBBR	SBSP	5998.0	1	873.332384

```
#Obtendo uma base de dados da empresa GOL
GLO = dados.groupby('Empresa').get_group('GLO')
```

In [ ]:

```
#Obtendo a tarifa máxima desta empresa
GLO[GLO.Tarifa == GLO.Tarifa.max()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
1349750	2022	4	GLO	SBRJ	SBRB	3676.43	1	3006.520999
1349752	2022	4	GLO	SBSP	SBRB	3676.43	1	2726.948456
1349754	2022	4	GLO	SBRB	SBMO	3676.43	3	3521.953075

In [ ]:

```
#Obtendo uma base de dados da empresa AZUL
AZU = dados.groupby('Empresa').get_group('AZU')
```

In [ ]:

```
#Obtendo a tarifa máxima desta empresa
AZU[AZU.Tarifa == AZU.Tarifa.max()]
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
1627127	2022	5	AZU	SSOU	SBCF	5965.9	1	376.064938

In [ ]:

```
#Obtendo a estatística descritiva de Tarifa por empresa
dados.groupby('Empresa').Tarifa.describe()
```

Out[ ]:

Empresa	count	mean	std	min	25%	50%	75%	max
ABJ	15.0	770.000000	336.367146	390.00	490.0	690.00	990.00	1490.00
AZU	897480.0	958.589508	594.872132	48.28	532.9	811.90	1241.90	5965.90
GLO	404842.0	953.553034	674.633136	21.00	444.9	758.90	1303.90	3676.43
PTB	4063.0	606.162771	327.968276	72.10	369.0	519.47	763.93	2470.69
TAM	564723.0	745.965793	585.803777	40.04	336.0	581.89	965.89	5998.00

```
#Obtendo uma base reduzida com empresas de interesse
empresas = ['AZU','GLO','TAM']
dados1 = dados[ dados['Empresa'].isin(empresas) ]
dados1.head()
```

Out[ ]:

	Ano	Mes	Empresa	Origem	Destino	Tarifa	Assentos	Distancia
0	2022	1	AZU	SBAC	SBAR	552.9	1	718.836149
1	2022	1	AZU	SBAC	SBBR	614.9	1	1675.186803
2	2022	1	AZU	SBAC	SBCA	550.9	1	2826.784854
3	2022	1	AZU	SBAC	SBCF	887.9	1	1804.621127
4	2022	1	AZU	SBAC	SBFL	438.9	3	2812.788957

In [ ]:

```
#Verificando a estatística do subgrupo em relação à distância
dados1.groupby('Empresa').Distancia.describe()
```

Out[ ]:

	count	mean	std	min	25%	50%	75
Empresa							
AZU	897480.0	1349.113501	792.378483	65.671788	676.552951	1187.064493	1948.74476
GLO	404842.0	1444.235669	800.774208	88.663432	798.603880	1315.103041	2077.38971
TAM	564723.0	1561.775178	809.735790	74.428631	866.562568	1482.018047	2265.31795

Visualização com Pandas

In [ ]:

```
dados2 = dados1[['Origem','Empresa','Tarifa','Distancia']]
origens = ['SBBR','SBGR','SBGL','SBCF']

dados2 = dados2[ dados2['Origem'].isin(origens) ]

dados2.head()
```

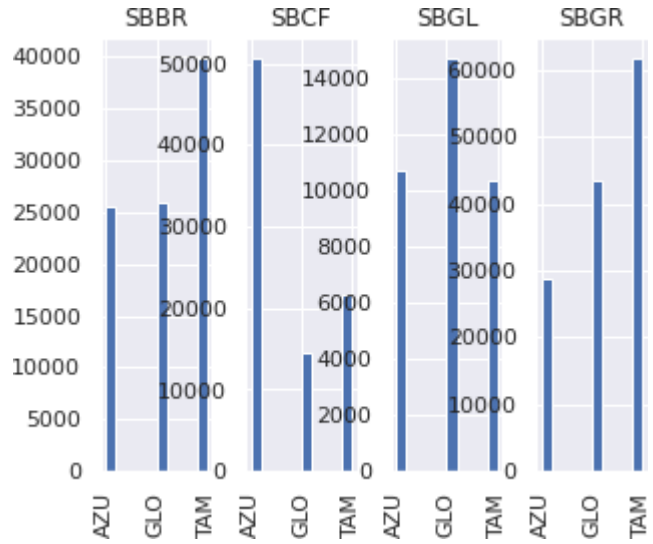
Out[ ]:

	Origem	Empresa	Tarifa	Distancia
6250	SBBR	AZU	1002.9	710.198748
6251	SBBR	AZU	1097.9	710.198748
6252	SBBR	AZU	1141.9	710.198748
6253	SBBR	AZU	1144.9	710.198748
6254	SBBR	AZU	1255.9	710.198748

```
dados2['Empresa'].hist(by=dados2['Origem']);
```

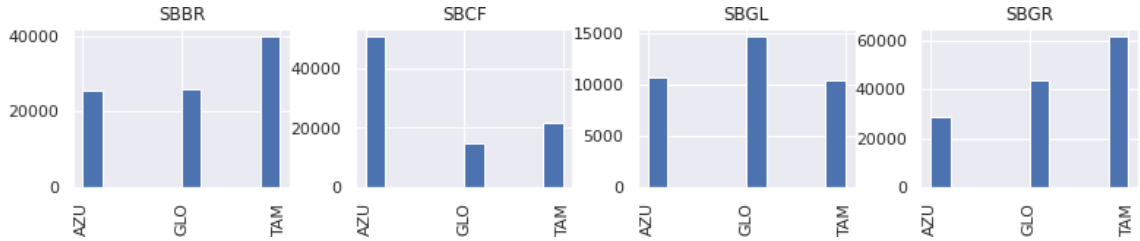
In [ ]:

```
dados2['Empresa'].hist(by=dados2['Origem'], layout=(1,5));
```



In [ ]:

```
dados2['Empresa'].hist(by=dados2['Origem'], layout=(1,5), figsize=(16,2));
```

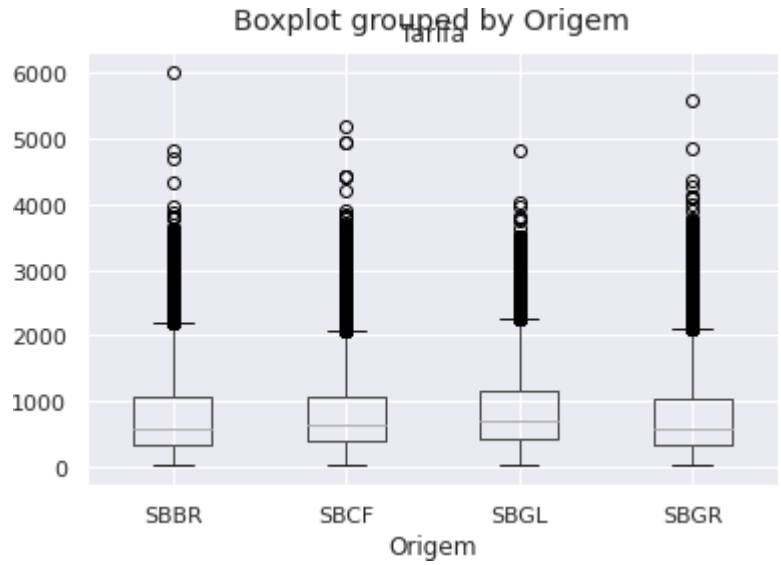


```
from numpy.core.memmap import dtype
dados2.boxplot(column='Tarifa', by='Origem');
'dtype=object'
```

```
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/_init_.py:1376:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequence
s (which is a list-or-tuple of lists-or-tuples-or ndarrays with different
lengths or shapes) is deprecated. If you meant to do this, you must specif
y 'dtype=object' when creating the ndarray.
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```

Out[ ]:

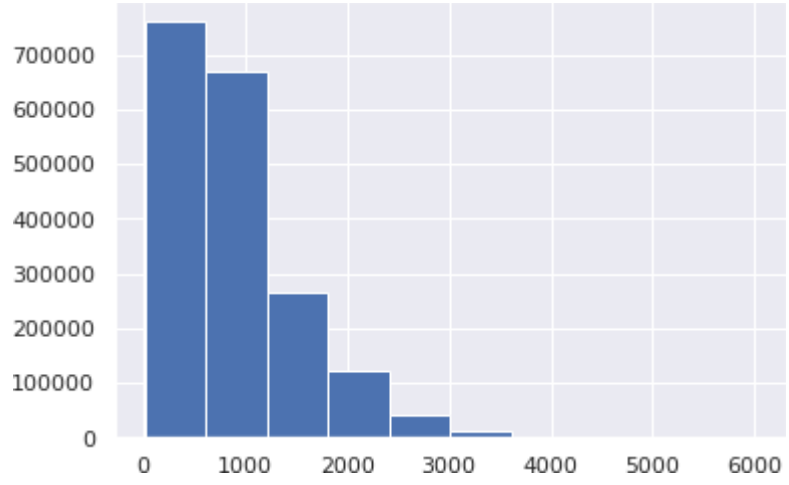
```
'dtype=object'
```



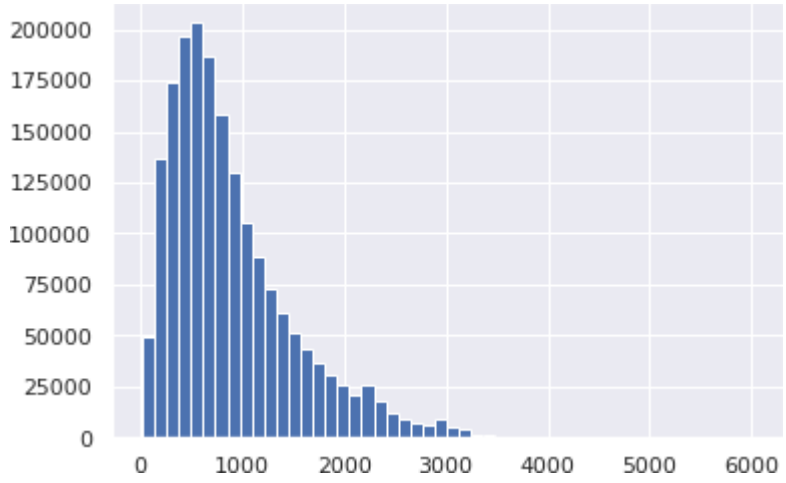
Visualização com Matplotlib

In [ ]:

```
plt.hist(dados['Tarifa'], bins=10)
plt.show()
```

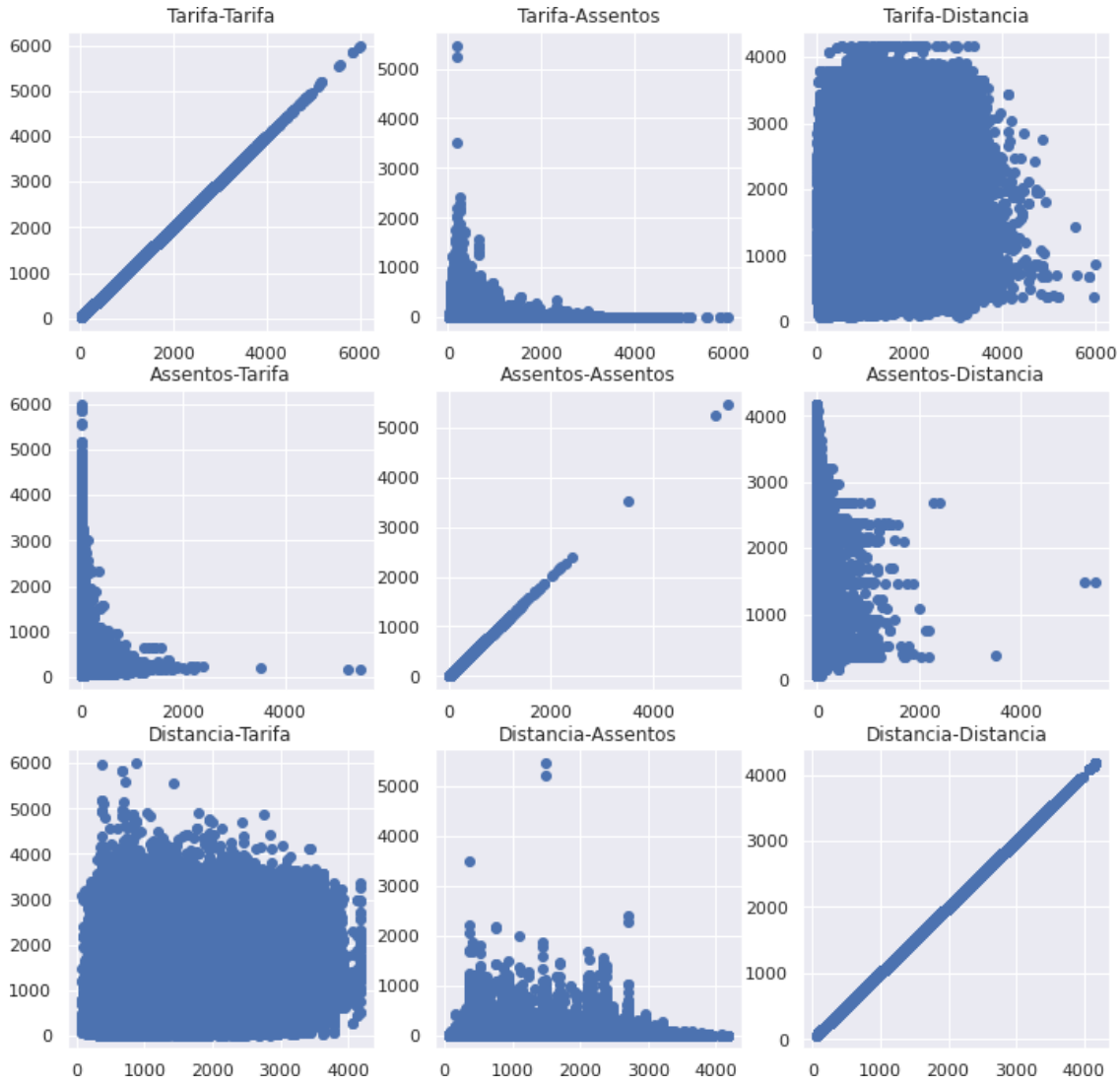


```
fig, ax = plt.subplots()
ax.hist(dados['Tarifa'], bins=50)
plt.show()
```



```
columns = ['Tarifa', 'Assentos', 'Distancia']

fig, ax = plt.subplots(3,3, figsize=(12,12))
for i in range(3):
    col1 = columns[i]
    for j in range(3):
        col2 = columns[j]
        ax[i][j].set_title(f'{col1}-{col2}')
        ax[i][j].scatter(dados[col1], dados[col2])
plt.show()
```


Tarifa-Tarifa
Tarifa-Assentos
Tarifa-Distancia
Assentos-Tarifa
Assentos-Assentos
Assentos-Distancia
Distancia-Tarifa
Distancia-Assentos
Distancia-Distancia

```
fig.savefig('correlations')
```

In [ ]:

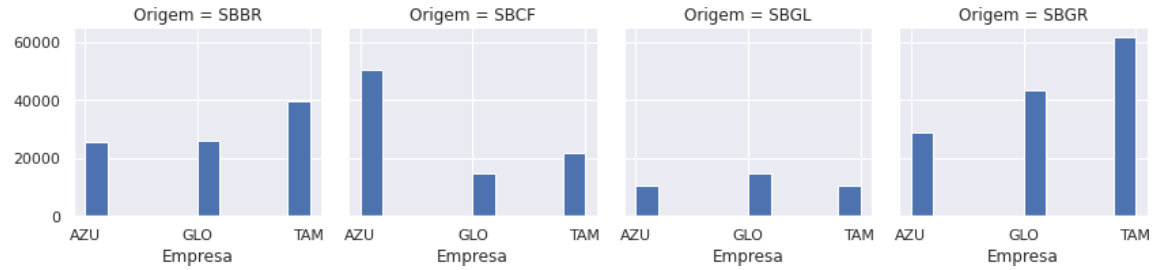
```
fig.show()
```

Visualização com Seaborn

Histogramas

In [ ]:

```
g = sns.FacetGrid(dados2, col="Origem")
g.map(plt.hist, "Empresa");
```

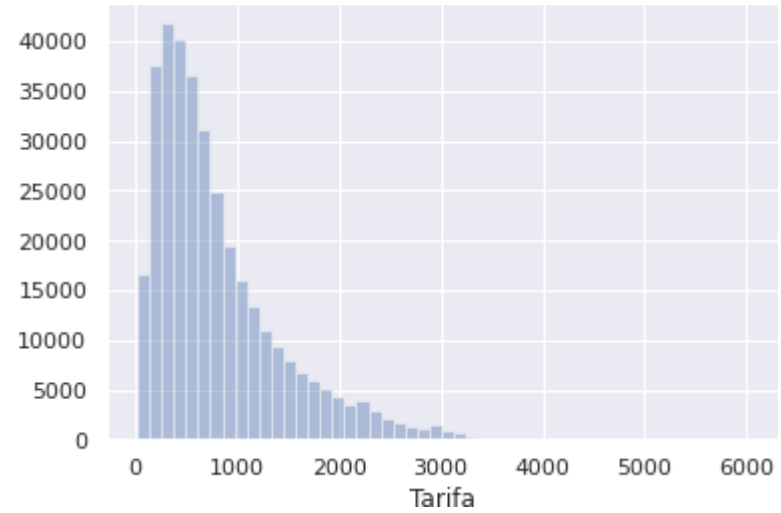


In [ ]:

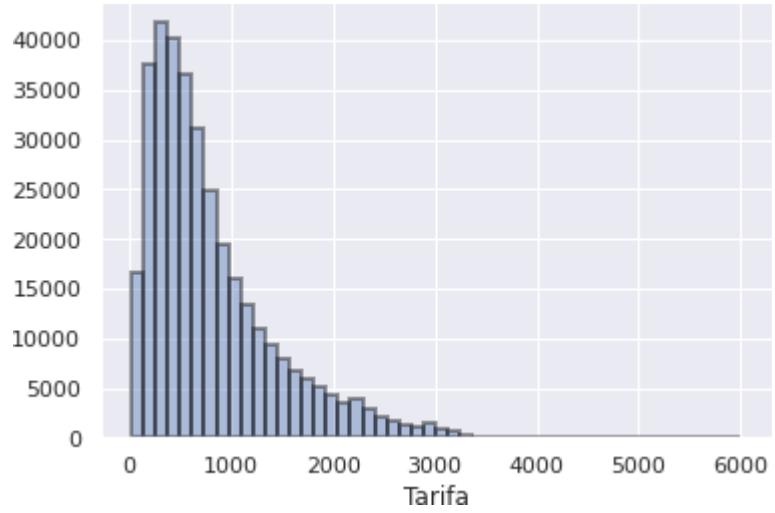
```
#https://seaborn.pydata.org/generated/seaborn.displot.html
#kde = kernel distribution estimation

sns.distplot(dados2.Tarifa,
             kde=False);
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: Futu
reWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-lev
el function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
```

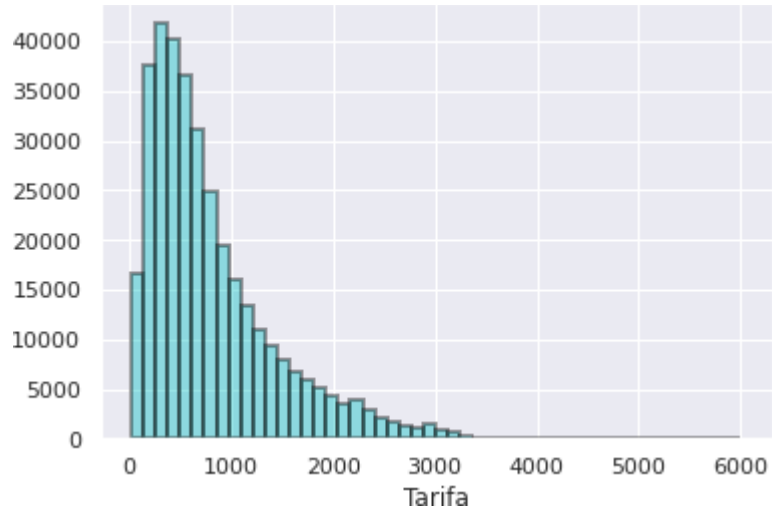


```
sns.distplot(dados2.Tarifa,
             kde=False,
             hist_kws=dict(edgecolor="black", linewidth=2));
```



In [ ]:

```
sns.distplot(dados2.Tarifa,
             kde=False,
             hist_kws=dict(edgecolor="black", linewidth=2),
             color='#00BFC4');
```

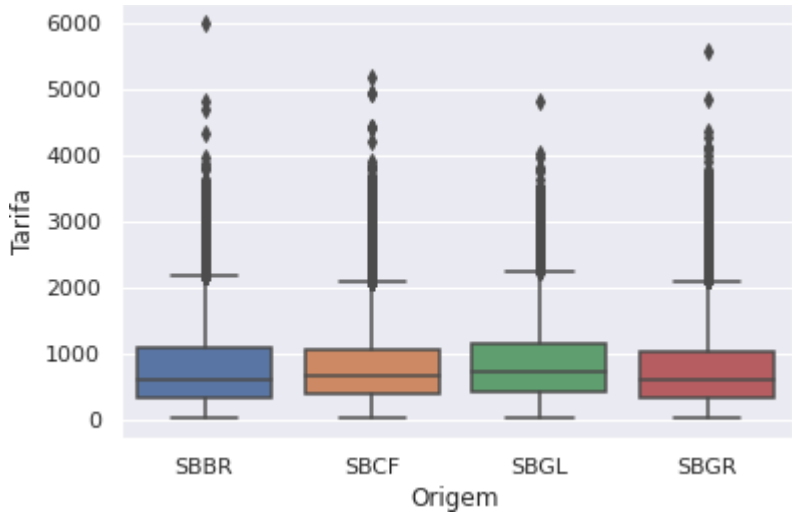


Boxplot

```
sns.boxplot(x = 'Origem', y = 'Tarifa', data = dados2)
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f53744fe650>
```

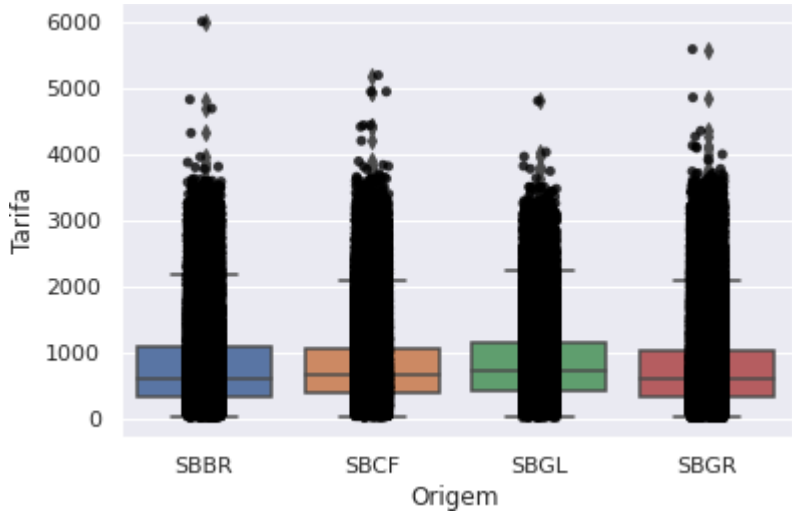


In [ ]:

```
sns.boxplot(x = 'Origem', y = 'Tarifa', data = dados2)
sns.stripplot(x = 'Origem', y = 'Tarifa', data = dados2,
              jitter=True,
              marker='o',
              alpha=0.8,
              color="black")
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5376e6a2d0>
```



Mapa de Calor

```
#Definindo a matriz de correlação
corr = dados2.corr()
corr
```

Out[ ]:

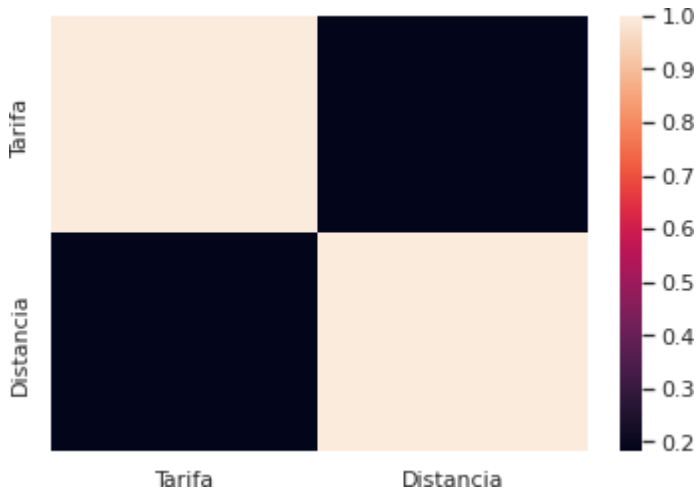
	Tarifa	Distancia
Tarifa	1.00000	0.18154
Distancia	0.18154	1.00000

In [ ]:

```
sns.heatmap(corr)
```

Out[ ]:

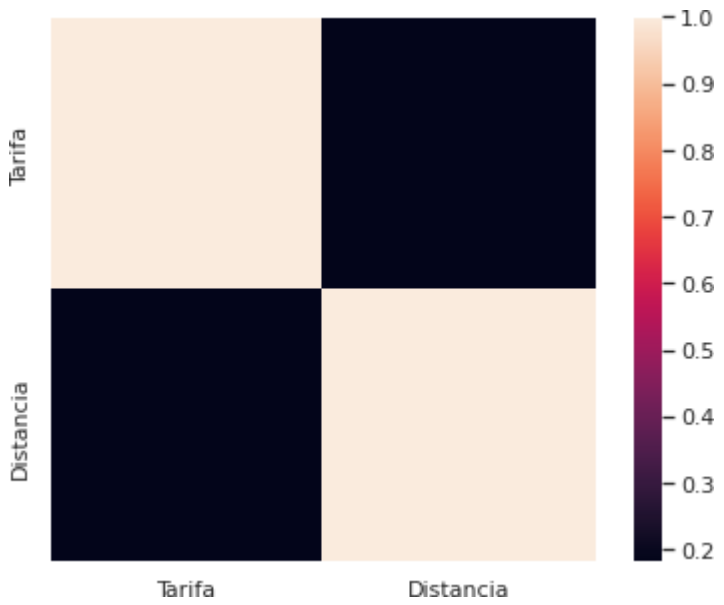
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5377eb19d0>
```



```
fig, ax = plt.subplots(figsize=(7,5))
sns.heatmap(corr, square=True)
```

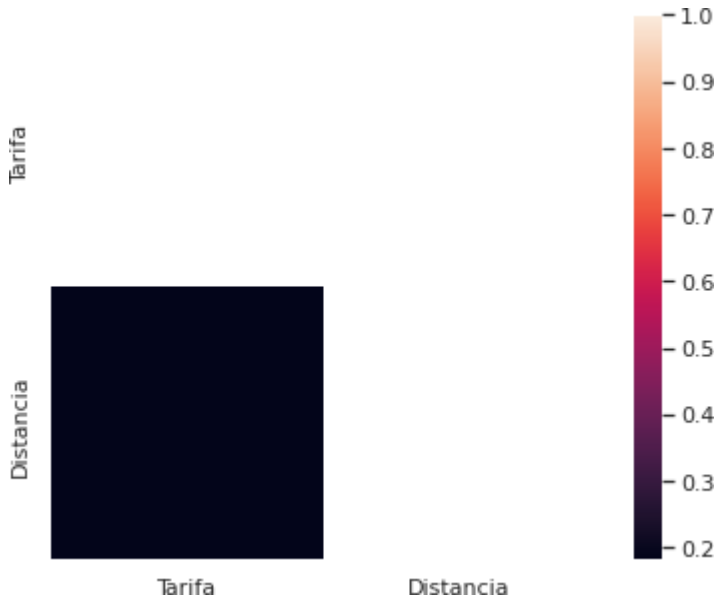
Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5377eee110>
```



```
# Mascarando a diagonal do mapa de calor
# https://seaborn.pydata.org/generated/seaborn.heatmap.html

mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)] = True
with sns.axes_style("white"):
    f, ax = plt.subplots(figsize=(7, 5))
    ax = sns.heatmap(corr, mask=mask, vmax=1, square=True)
```



Scatter Plot

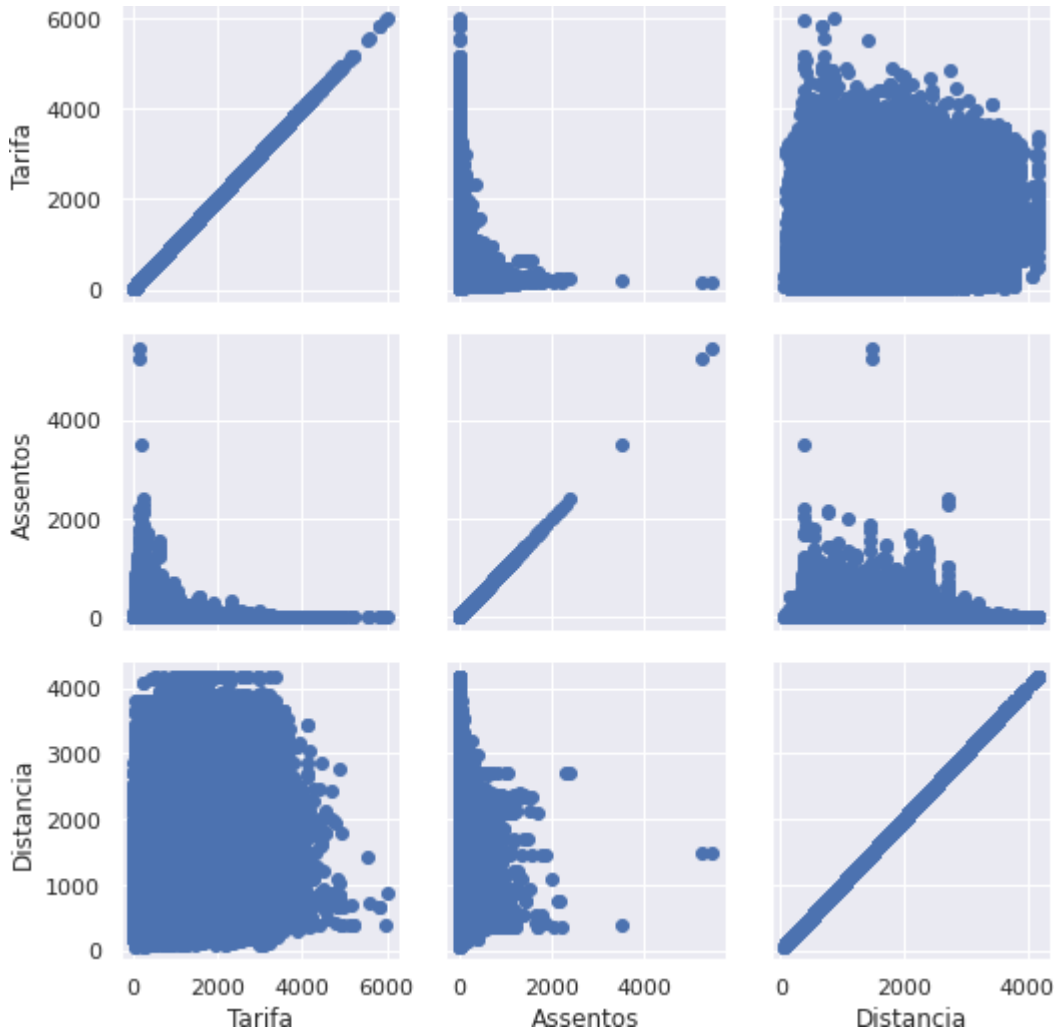
In [ ]:

```
#Seleciona colunas com dados numéricos
number = dados.select_dtypes(include=['number'])
```

In [ ]:

```
#OU seleciona manualmente as colunas de interesse, com dados numéricos
colunas = ['Tarifa', 'Assentos', 'Distancia']
dados3 = dados[colunas]
```

```
g = sns.PairGrid(dados3)
g.map(plt.scatter);
```



ANEXOS

ANEXO I – Lei de Criação da ANAC nº 11.182, de 27 de setembro de 2005

ANEXO II – Decreto de Instalação da ANAC nº 5.731, de 20 de março de 2006

ANEXO III – Regimento Interno: Resolução nº 381, de 14 de junho de 2016

ANEXO IV – Alteração do RI: Resolução nº 581, de 21 de agosto de 2020

ANEXO V – Instrução Normativa nº 127, de 5 de outubro de 2018