Aula 9 – Tratamento de Dados Ausentes (Missing Values)

Desvendando os Buracos nos Dados: A Arte de Lidar com Valores Ausentes

Bem-vindo à Aula 9 do nosso Curso de Análise Exploratória de Dados! Imagine que você está montando um quebra-cabeça complexo, mas algumas peças simplesmente não estão lá. Ou, talvez, você esteja preenchendo um formulário importante e algumas perguntas são deixadas em branco. No mundo dos dados, essas "peças ausentes" ou "campos em branco" são o que chamamos de **dados ausentes** ou *missing values*. Eles são um dos desafios mais comuns e, muitas vezes, mais frustrantes que você enfrentará ao trabalhar com dados reais.

Nesta aula, nosso objetivo é equipar você com as ferramentas e o raciocínio necessários para enfrentar esse desafio de frente. Ao final desta aula, você será capaz de identificar, compreender e aplicar as principais estratégias para lidar com dados ausentes, garantindo que suas análises sejam robustas e confiáveis. Vamos explorar desde a detecção desses "buracos" até as diferentes abordagens para preenchê-los ou removê-los, sempre com um olhar prático e focado nas ferramentas que o mercado de trabalho exige.

A relevância deste tópico não pode ser subestimada. Dados ausentes podem distorcer resultados, invalidar modelos e levar a conclusões errôneas, impactando desde a avaliação de um projeto universitário até a tomada de decisões estratégicas em uma empresa ou a precisão de um algoritmo em um concurso público. Dominar o tratamento de dados ausentes é um passo crucial para se tornar um analista de dados competente e confiável.

Nossa jornada começará com a identificação desses valores, usando as poderosas bibliotecas Python como Pandas. Em seguida, mergulharemos nas estratégias de tratamento, ponderando entre a remoção e o preenchimento (imputação). Por fim, exploraremos os métodos de imputação mais comuns e como visualizar a distribuição desses dados ausentes para tomar decisões mais informadas. Prepare-se para transformar dados imperfeitos em insights valiosos!

O Problema Silencioso: Compreendendo os Dados Ausentes

Você já parou para pensar por que alguns dados simplesmente não aparecem onde deveriam? Imagine que você está coletando informações para um projeto de pesquisa universitário sobre hábitos de estudo, e alguns alunos deixam a pergunta sobre "horas de sono" em branco. Ou, em um cenário de concurso público, um candidato esquece de preencher o campo de "experiência profissional" em um formulário online. Essas lacunas, que parecem pequenas, podem ter um impacto gigantesco na qualidade e na confiabilidade da sua análise.

> **Importante:** Os dados ausentes não são apenas "espaços em branco"; eles são um problema que pode minar a integridade de qualquer conjunto de dados.

Eles podem surgir por diversas razões: um erro humano durante a entrada de dados, falhas em sensores ou sistemas de coleta, a não aplicabilidade de uma pergunta para um determinado indivíduo (por exemplo, "número de filhos" para alguém sem filhos), ou até mesmo a recusa de um participante em fornecer uma informação. Ignorá-los é como construir uma casa com tijolos faltando: a estrutura pode parecer sólida por fora, mas está comprometida por dentro.

NaN (Not a Number)

Padrão para valores numéricos ausentes

None

Representação para objetos ausentes

Distinção Crucial

NaN ≠ zero ≠ string vazia

No universo da programação e da análise de dados, especialmente com Python e a biblioteca Pandas, os dados ausentes são frequentemente representados de formas específicas. O mais comum é o NaN (Not a Number), que é o padrão para valores numéricos ausentes, ou None para objetos. É fundamental entender que NaN não é zero, nem uma string vazia; é uma representação de "não há valor aqui". Essa distinção é crucial, pois tratá-los incorretamente pode levar a cálculos errados e análises distorcidas.

Pense nos dados ausentes como peças de um quebra-cabeça que se perderam. Se você tentar montar o quebra-cabeça sem essas peças, o resultado final estará incompleto e talvez até distorcido. Da mesma forma, se você analisar um conjunto de dados com lacunas sem tratá-las adequadamente, suas conclusões podem não refletir a realidade, comprometendo a validade do seu trabalho, seja ele um TCC ou uma análise para um edital.

Detectando as Lacunas: Seu Primeiro Passo Crucial

Antes de pensar em como resolver um problema, você precisa saber onde ele está, certo? No tratamento de dados ausentes, a lógica é a mesma. Não adianta tentar preencher ou remover valores se você não consegue identificá-los de forma eficiente. Em conjuntos de dados pequenos, talvez você consiga ver as lacunas a olho nu, mas e se você estiver lidando com milhares ou milhões de linhas e centenas de colunas? É aí que a programação se torna sua melhor amiga.

01

Função .isnull()

Sua lupa digital que retorna True onde há valores ausentes e False onde há dados presentes

02

Combinação com .sum()

Soma todos os True para contar o total de valores ausentes por coluna

03

Panorama Completo

Visão quantitativa da extensão do problema em cada variável

A biblioteca Pandas, que é o padrão da indústria para manipulação de dados em Python, oferece ferramentas poderosas e intuitivas para essa tarefa. A função .isnull() é a sua lupa digital. Quando aplicada a um DataFrame, ela retorna outro DataFrame do mesmo tamanho, mas preenchido com True onde há um valor ausente e False onde o valor está presente. É como ter um mapa que destaca exatamente onde estão os "buracos" no seu conjunto de dados.

Mas ver um DataFrame cheio de True e False ainda pode ser esmagador. Para ter uma visão mais concisa e quantitativa, combinamos .isnull() com .sum(). Ao encadear essas duas funções (df.isnull().sum()), o Pandas soma todos os True (que são interpretados como 1) para cada coluna, revelando o número total de valores ausentes em cada uma delas. Isso nos dá um panorama rápido e preciso da extensão do problema, permitindo priorizar quais colunas precisam de mais atenção.

```
# Exemplo de saída:
Nome        0
Idade       3
Nota        0
Frequencia  1
dtype: int64
```

Vamos a um exemplo prático. Imagine que você tem um DataFrame df com informações de alunos, incluindo Nome, Idade, Nota e Frequencia. Se alguns alunos não informaram a idade ou a frequência, o código df.isnull().sum() rapidamente mostraria a saída acima. Essa saída indica que há 3 valores ausentes na coluna 'Idade' e 1 na coluna 'Frequencia'. Essa é a primeira e mais vital etapa em qualquer processo de limpeza de dados. Sem essa detecção precisa, qualquer estratégia de tratamento seria um tiro no escuro.

Remover ou Preencher? O Dilema Central do Tratamento

Uma vez que você identificou os dados ausentes, a próxima grande questão surge: o que fazer com eles? Essa é a encruzilhada mais importante no tratamento de dados. Não existe uma resposta única e universal; a melhor abordagem depende do contexto, da quantidade de dados ausentes e da natureza da sua análise. É como decidir o que fazer com uma rachadura na parede da sua casa: você a remove (demolindo a parede) ou a preenche (com massa corrida)? Ambas as opções têm suas vantagens e desvantagens.

Estratégia 1: Remoção

A primeira estratégia é a **remoção**. Isso significa simplesmente descartar as linhas (registros) ou colunas (variáveis) que contêm valores ausentes. Parece simples, não é? E, de fato, é a abordagem mais direta. Se a quantidade de dados ausentes for muito pequena em relação ao tamanho total do seu conjunto de dados, ou se a ausência de um valor em uma linha específica torna essa linha inútil para sua análise, a remoção pode ser a solução mais rápida e eficaz.

Risco: O grande risco aqui é a perda de informações valiosas. Se você remover muitas linhas, seu conjunto de dados pode encolher drasticamente, comprometendo a representatividade e o poder estatístico da sua análise.

Estratégia 2: Preenchimento

A segunda estratégia é o **preenchimento**, também conhecido como **imputação**. Em vez de descartar os dados, você tenta estimar e substituir os valores ausentes por outros valores. Pense nisso como preencher os espaços em branco de um formulário com as informações mais prováveis, baseadas no que você já sabe.

Desafio: O desafio, porém, é escolher o método de imputação correto, pois um preenchimento inadequado pode introduzir viés ou distorcer a variabilidade dos seus dados, levando a conclusões falsas.

Conceito	Âmbito/Aplicação	Base/Origem	Exemplo
Remoção	Quando a proporção de dados ausentes é pequena	Descarte de linhas/colunas incompletas	df.dropna() para remover linhas com qualquer valor ausente
Imputação	Quando a perda de dados é inaceitável	Estimativa baseada em dados existentes	df['coluna'].fillna(df['coluna'].mean()) para preencher com a média

A escolha entre remover e imputar é uma decisão crítica que exige reflexão. Não se trata apenas de aplicar uma função, mas de entender as implicações estatísticas e práticas de cada abordagem para o seu projeto.

Estratégia 1: A "Limpeza Geral" – Removendo Dados Ausentes

Às vezes, a solução mais simples é a mais eficaz. Quando nos deparamos com dados ausentes, uma das primeiras abordagens que podemos considerar é a remoção. Pense em um armário bagunçado: se há algumas peças de roupa que estão velhas, rasgadas e não servem mais, a melhor opção pode ser simplesmente jogá-las fora para liberar espaço e organizar o que realmente importa. No contexto dos dados, isso significa descartar as linhas (registros) ou colunas (variáveis) que contêm os valores ausentes.

Função .dropna()

Ferramenta principal para remoção de dados ausentes no Pandas

- Remove linhas ou colunas com valores ausentes
- Implementação direta e fácil

Controle Flexível

Parâmetros para personalizar a remoção

- axis=1 para remover colunas
- thresh=3 para manter linhas com pelo menos 3 valores válidos

A remoção é uma estratégia direta e fácil de implementar, especialmente com o Pandas. A função .dropna() é a sua ferramenta para isso. Ela permite que você remova linhas ou colunas inteiras que contenham pelo menos um valor ausente. Por exemplo, se uma pesquisa tem uma linha inteira de respostas em branco, ou se uma coluna de dados de sensor está quase toda vazia, removê-las pode ser a decisão mais sensata.

> **Cuidado:** O principal risco da remoção é a perda de informações. A remoção é mais indicada quando a proporção de dados ausentes é muito pequena (geralmente menos de 5% em uma coluna específica).

No entanto, é crucial usar essa abordagem com cautela. O principal risco da remoção é a perda de informações. Se você tem um conjunto de dados pequeno e remove muitas linhas, pode acabar com um conjunto de dados tão reduzido que ele não é mais representativo da população original, ou que não possui dados suficientes para uma análise estatística robusta. Isso é especialmente problemático em concursos públicos, onde a precisão e a completude dos dados podem ser decisivas. A remoção é mais indicada quando a proporção de dados ausentes é muito pequena (geralmente menos de 5% em uma coluna específica) e a ausência é considerada "Missing Completely At Random" (MCAR), ou seja, a ausência não está relacionada a nenhum outro valor no conjunto de dados.

```
# Exemplos práticos:
df.dropna()               # Remove linhas com qualquer valor ausente
df.dropna(axis=1)         # Remove colunas com valores ausentes
df.dropna(thresh=3)       # Remove linhas com menos de 3 valores válidos
```

Vamos ver como funciona na prática. Se você tem um DataFrame df e quer remover todas as linhas que contenham *qualquer* valor ausente, você usaria df.dropna(). Se preferir remover apenas as colunas que têm valores ausentes, o comando seria df.dropna(axis=1). É possível até especificar um limiar de valores não-ausentes (thresh) para manter linhas ou colunas que tenham um mínimo de dados válidos. Por exemplo, df.dropna(thresh=3) removeria linhas que têm menos de 3 valores não-ausentes. Essa flexibilidade permite um controle mais refinado sobre o processo de limpeza.

Estratégia 2: O "Preenchimento Inteligente" – Imputação de Dados Ausentes

Remover dados é como jogar fora peças de um quebra-cabeça que não se encaixam perfeitamente. Mas e se essas peças, mesmo com defeito, forem cruciais para a imagem final? É aí que entra a **imputação**, a estratégia de preencher os valores ausentes com estimativas. Em vez de descartar informações, tentamos inferir qual seria o valor mais provável, baseando-nos nos dados que já temos. É como preencher um formulário com base no histórico da pessoa, em vez de simplesmente descartar o formulário porque um campo ficou em branco.

Preserva o Tamanho

Mantém o poder estatístico e a representatividade da amostra

Flexibilidade

Diversos métodos disponíveis, desde simples até complexos

Requer Cuidado

Preenchimento inadequado pode introduzir viés nos dados

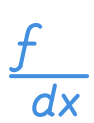
A imputação é uma técnica poderosa porque permite preservar o tamanho do seu conjunto de dados, o que é vital para manter o poder estatístico e a representatividade da sua amostra. Imagine que você está analisando dados de desempenho de alunos para um projeto e percebe que alguns alunos não têm a nota de uma prova específica. Se você simplesmente remover esses alunos, pode perder informações valiosas sobre o desempenho geral da turma ou sobre fatores que influenciam o aprendizado. A imputação oferece uma alternativa, permitindo que você mantenha esses alunos na análise, preenchendo suas notas ausentes com uma estimativa razoável.

No entanto, a imputação não é uma bala de prata. Ela exige um entendimento cuidadoso dos seus dados e do impacto que cada método de preenchimento pode ter. Um preenchimento inadequado pode introduzir viés, reduzir a variabilidade natural dos dados ou até mesmo criar relações espúrias que não existem na realidade. Por isso, a escolha do método de imputação é uma das decisões mais importantes no processo de tratamento de dados ausentes.

A beleza da imputação reside na sua flexibilidade. Existem diversos métodos, desde os mais simples e diretos até os mais complexos e baseados em modelos estatísticos avançados. A escolha dependerá do tipo de dado (numérico ou categórico), da distribuição da variável e da natureza da ausência. Nas próximas páginas, exploraremos os métodos de imputação mais comuns e práticos, que você poderá aplicar imediatamente em seus projetos e análises.

Métodos de Imputação: As Abordagens "Médias" (Média, Mediana, Moda)

Quando decidimos preencher os dados ausentes, a pergunta que surge é: com o quê? A forma mais intuitiva de preencher uma lacuna é usar um valor que seja "típico" ou "representativo" dos dados existentes. Pense em um grupo de amigos que sempre pede pizza. Se um deles não consegue decidir o sabor, o grupo pode sugerir o sabor mais pedido (moda), o sabor que está no meio da lista de preferências (mediana), ou uma média de todos os sabores já pedidos (se pudéssemos quantificar o "sabor"). No mundo dos dados, essas "médias" são a média, a mediana e a moda.



Média

Para dados numéricos com distribuição simétrica. Soma todos os valores e divide pelo número de observações. **Cuidado:** Muito sensível a outliers (valores extremos).

```
df['idade'].fillna(df['idade'].mean(), inplace=True)
```

Mediana

Valor central dos dados ordenados. Robusta para distribuições assimétricas e com outliers. Representa bem o "valor típico" da maioria dos dados.

```
df['salario'].fillna(df['salario'].median(), inplace=True)
```

Moda

Valor mais frequente. Ideal para dados categóricos ou numéricos com muitos valores repetidos. Garante que o valor imputado seja válido e comum.

```
df['estado_civil'].fillna(df['estado_civil'].mode()[0], inplace=True)
```

Para dados numéricos, a **média** é frequentemente a primeira opção que vem à mente. Ela é calculada somando todos os valores existentes em uma coluna e dividindo pelo número de valores. Preencher com a média é simples e mantém a média geral da coluna inalterada. No entanto, ela é muito sensível a **outliers** (valores extremos). Se sua coluna tiver alguns valores muito altos ou muito baixos, a média pode ser distorcida e, consequentemente, o valor imputado pode não ser tão representativo.

A **mediana** é uma alternativa robusta para dados numéricos, especialmente quando há outliers ou a distribuição dos dados é assimétrica (enviesada). A mediana é o valor central em um conjunto de dados ordenado. Metade dos valores está acima dela e metade está abaixo. Preencher com a mediana é como escolher o "meio-termo" do grupo, sendo menos afetada por valores extremos. Por exemplo, se você tem salários e alguns CEOs ganham milhões, a média salarial seria inflacionada, mas a mediana ainda representaria bem o salário típico da maioria.

Para dados categóricos (como "cor favorita", "estado civil") ou numéricos com muitos valores repetidos, a **moda** é a escolha ideal. A moda é o valor que aparece com maior frequência em uma coluna. Preencher com a moda é como escolher o sabor de pizza que a maioria dos seus amigos prefere. É uma abordagem simples e lógica para dados não numéricos, garantindo que o valor imputado seja um dos valores válidos e mais comuns na categoria.

> **Dica Prática:** Analisar a distribuição dos dados (com histogramas, por exemplo) antes de imputar é uma prática recomendada para escolher o método mais adequado.

Métodos de Imputação: Abordagens "Constantes" e "Avançadas"

Nem sempre a média, mediana ou moda são as melhores opções para preencher dados ausentes. Em alguns cenários, a ausência de um valor pode carregar um significado próprio, ou os dados podem exigir uma abordagem mais sofisticada. É como em um formulário de inscrição para um concurso: se o campo "número de dependentes" está em branco, pode significar que a pessoa não tem dependentes, e preencher com a média de dependentes de outros candidatos seria incorreto. Nesses casos, a imputação com um **valor constante** se torna uma alternativa valiosa.

Imputação Constante

Preencher com um valor constante significa substituir todos os valores ausentes em uma coluna por um número ou uma string específica que você define. Por exemplo, se a ausência de um valor na coluna renda_extra significa que a pessoa não tem renda extra, você pode preencher os NaNs com 0. Ou, se em uma pesquisa de opinião, a ausência de resposta em uma pergunta indica "não se aplica" ou "não sei", você pode preencher com a string 'Não Informado' ou 'Desconhecido'.

```
# Exemplos práticos:
df['renda_extra'].fillna(0)
df['opiniao'].fillna('Não Informado')
```

Vantagens

- Simples e transparente
- Evita introdução de viés estatístico
- Adequado quando ausência tem significado específico
- Preserva a interpretabilidade dos dados

Técnicas Avançadas de Imputação

Interpolação

Preenche valores ausentes com base nos valores vizinhos, especialmente útil para séries temporais onde há uma sequência lógica nos dados.

K-Nearest Neighbors (KNN) Imputer

Preenche valores ausentes usando os valores dos "vizinhos" mais próximos no conjunto de dados, baseado em similaridade.

Multiple Imputation by Chained Equations (MICE)

Método iterativo que modela cada variável com dados ausentes como uma função de outras variáveis no conjunto de dados.

Além dos métodos básicos (média, mediana, moda, constante), o campo da ciência de dados oferece técnicas de imputação mais **avançadas**. Essas técnicas são geralmente mais complexas e exigem um conhecimento mais aprofundado de estatística e machine learning, mas podem oferecer resultados superiores, especialmente quando os dados ausentes não são aleatórios e há padrões complexos a serem inferidos.

Conceito	Âmbito/Aplicação	Base/Origem	Exemplo
Constante	Quando a ausência tem um significado específico	Valor fixo definido pelo analista	df['renda_extra'].fillna(0)
Avançadas	Quando há padrões complexos de ausência	Modelos estatísticos/Machine Learning	IterativeImputer (Scikit-learn)

Essas técnicas avançadas são tópicos para estudos futuros, mas é importante saber que elas existem e são amplamente utilizadas em cenários mais complexos. Para a maioria dos casos práticos e para o nível de um curso introdutório, os métodos de média, mediana, moda e constante são mais do que suficientes e representam a base do tratamento de dados ausentes.

Visualizando os Vazios: Ver para Crer nos Dados Ausentes

Números e tabelas são essenciais, mas o cérebro humano é incrivelmente bom em reconhecer padrões visuais. Quando se trata de dados ausentes, visualizar a extensão e a distribuição dessas lacunas pode revelar insights que uma simples contagem numérica jamais mostraria. É como olhar para um mapa de uma cidade: ver os buracos no mapa de ruas é muito mais impactante do que apenas ler uma lista de endereços ausentes. A visualização nos ajuda a entender se a ausência é aleatória, se está concentrada em certas colunas ou linhas, ou se há um padrão que pode indicar um problema subjacente na coleta de dados.

Heatmap com Seaborn

sns.heatmap(df.isnull()) cria um mapa de calor onde células ausentes são coloridas diferentemente

Gráfico de Barras

df.isnull().sum().plot(kind='bar') mostra contagem de valores ausentes por coluna

Identificação de Padrões

Revela correlações na ausência entre diferentes colunas e variáveis

A biblioteca Seaborn, construída sobre Matplotlib, é uma ferramenta fantástica para criar visualizações estatísticas atraentes e informativas. Uma das formas mais eficazes de visualizar dados ausentes é através de um **heatmap**. Ao aplicar sns.heatmap(df.isnull()), você gera um mapa de calor onde cada célula representa um ponto de dado. As células com valores ausentes são coloridas de uma forma (por exemplo, branco) e as células com dados presentes de outra (por exemplo, azul escuro). Isso permite que você veja rapidamente a densidade de valores ausentes em cada coluna e, mais importante, se há correlações na ausência entre diferentes colunas. Por exemplo, você pode notar que sempre que a Idade está ausente, o Salário também está.

Outra visualização útil é um simples **gráfico de barras** que mostra a contagem de valores ausentes por coluna. Isso pode ser feito facilmente com df.isnull().sum().plot(kind='bar'). Embora não revele padrões de correlação, ele oferece uma visão clara das colunas mais problemáticas em termos de volume de dados ausentes.

> **Biblioteca Adicional:** Para análises mais aprofundadas, a biblioteca `missingno` oferece visualizações específicas como matrizes de ausência e dendrogramas para identificar relações complexas entre variáveis.

Para análises mais aprofundadas de padrões de ausência, a biblioteca missingno (não faz parte do Pandas/Seaborn padrão, mas é muito útil) oferece visualizações específicas, como matrizes de ausência e dendrogramas, que podem ajudar a identificar relações complexas entre a ausência de dados em diferentes variáveis. A capacidade de "ver" os dados ausentes é um passo crucial para tomar decisões informadas sobre a melhor estratégia de tratamento. Ela complementa a análise numérica e adiciona uma camada de compreensão que é indispensável para uma análise de dados robusta e reprodutível, especialmente em ambientes como Jupyter Notebooks.

Consolidação e Próximos Passos

Chegamos ao fim de nossa jornada sobre o tratamento de dados ausentes, um dos pilares da limpeza e preparação de dados. Começamos entendendo o que são esses "buracos" em nossos conjuntos de dados e por que eles surgem. Em seguida, aprendemos a identificá-los de forma eficiente usando as funções .isnull() e .sum() do Pandas, transformando um problema invisível em algo quantificável.

01

Identificação

Detectar valores ausentes com .isnull() e .sum() do Pandas

02

Estratégia

Escolher entre remoção (.dropna()) e imputação (.fillna())

03

Método

Aplicar média, mediana, moda ou valores constantes conforme apropriado

04

Visualização

Usar heatmaps com Seaborn para identificar padrões de ausência

Exploramos as duas grandes estratégias: a remoção e a imputação. Vimos que a remoção, embora simples com .dropna(), deve ser usada com cautela para evitar a perda excessiva de informações. Por outro lado, a imputação, com métodos como média, mediana, moda e valores constantes, nos permite preencher as lacunas, preservando o tamanho do dataset, mas exigindo uma escolha cuidadosa para não introduzir viés. Finalmente, destacamos a importância da visualização, usando heatmaps com Seaborn, para entender os padrões de ausência e tomar decisões mais informadas.

Em prática:

- Sempre comece sua análise de dados verificando a presença de valores ausentes.
- Analise a proporção de dados ausentes e o tipo de variável antes de decidir entre remoção e imputação.
- Para dados numéricos, considere a média para distribuições simétricas e a mediana para distribuições enviesadas ou com outliers.
- Para dados categóricos, a moda ou um valor constante como "Desconhecido" são boas opções.
- Visualize os dados ausentes para identificar padrões e correlações que podem influenciar sua estratégia.

Autoavaliação

1 (Nível Fácil)

Qual função do Pandas é utilizada para identificar valores ausentes em um DataFrame?

- a) .notnull()
- b) .isna()
- c) .isnull()
- d) .hasnull()

2 (Nível Médio)

Em um conjunto de dados com uma coluna de salários que apresenta alguns valores ausentes e uma distribuição assimétrica (com alguns salários muito altos), qual método de imputação seria mais adequado para preencher esses valores?

- a) Média
- b) Mediana
- c) Moda
- d) Um valor constante (ex: 0)

3 (Nível Médio)

Qual é o principal risco de remover linhas com dados ausentes de um conjunto de dados?

- a) Aumento do tempo de processamento.
- b) Perda significativa de informações e redução do poder estatístico.
- c) Introdução de novos valores ausentes.
- d) Dificuldade em visualizar os dados restantes.

(Nível Difícil)

Você está analisando dados de um censo e percebe que a coluna 'Renda Familiar' possui 40% de valores ausentes. Além disso, você suspeita que a ausência de dados nessa coluna está relacionada ao nível de escolaridade dos entrevistados. Qual seria a abordagem mais prudente para lidar com essa situação?

- a) Remover a coluna 'Renda Familiar' inteira.
- b) Preencher os valores ausentes com a média da 'Renda Familiar'.
- c) Utilizar um método de imputação mais avançado, como MICE ou KNN, após investigar a relação com a escolaridade.
- d) Preencher os valores ausentes com 0, assumindo que a ausência significa "sem renda".

(Questão Discursiva)

Explique, com suas palavras, a diferença entre a imputação por média e por mediana, e em que tipo de situação cada uma seria mais recomendada.

Gabarito

1. c)

A função .isnull() é utilizada para identificar valores ausentes

2. b)

Mediana é mais adequada para distribuições assimétricas

3. b)

Perda significativa de informações é o principal risco

4. c)

Métodos avançados são necessários para padrões complexos

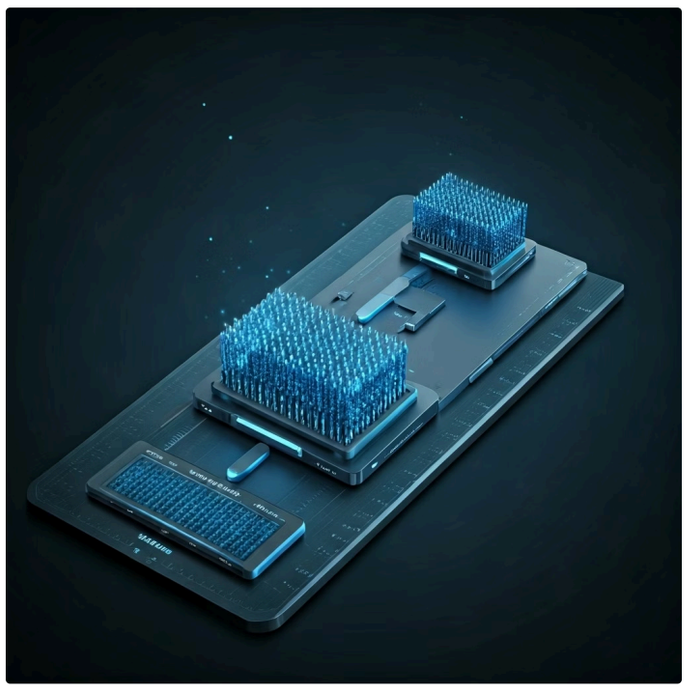
5. Resposta Discursiva:

> A imputação por **média** substitui os valores ausentes pelo valor médio de todos os dados existentes na coluna. É recomendada para distribuições de dados numéricos que são simétricas e não possuem muitos outliers, pois a média é sensível a valores extremos. Já a imputação por **mediana** substitui os valores ausentes pelo valor central dos dados ordenados na coluna. É mais robusta e recomendada para distribuições assimétricas ou quando há a presença de outliers, pois a mediana é menos afetada por esses valores extremos, representando melhor o "valor típico" da maioria dos dados.

Próxima Aula e Recursos Adicionais

Próxima Aula:

Na Aula 10, daremos um passo adiante na preparação de dados, explorando a **Transformação e Criação de Novas Colunas**. Você aprenderá a derivar novas informações a partir dos dados existentes, enriquecendo seu conjunto de dados para análises mais profundas e modelos mais poderosos.



Recursos Adicionais:

- **Documentação Pandas:** Para explorar a fundo as funções .isnull(), .dropna() e .fillna().
- **Documentação Seaborn:** Para aprofundar nas visualizações de dados, incluindo heatmaps.
- **Kaggle Datasets:** Para praticar o tratamento de dados ausentes em conjuntos de dados reais e desafiadores.

NOTA IMPORTANTE: As informações técnicas desta aula estão atualizadas até 2025. Consulte sempre fontes oficiais e a documentação das bibliotecas para verificar alterações e novas funcionalidades.