



Agrupamento, Agregação e Criação de Indicadores Chave (KPIs)

Transformando Dados em Decisões Estratégicas

Aula 15 - UC2 | Ciência de Dados

Objetivos da Aula



Dominar Agrupamento e Agregação

Compreender e aplicar o comando `groupby()` e funções de agregação (`mean`, `sum`, `count`, `max`, `min`) para transformar dados brutos em informações resumidas e acionáveis.



Criar KPIs Acionáveis

Desenvolver a capacidade de criar Indicadores Chave de Desempenho (KPIs) que medem o sucesso e orientam decisões estratégicas de negócio, como Ticket Médio e Volume de Vendas.



Aplicar Filtragem Lógica

Utilizar operadores lógicos (`&` e `|`) para refinar dados antes de calcular KPIs, permitindo análises específicas para subconjuntos de dados e segmentações estratégicas.



Documentar Scripts Profissionalmente

Desenvolver boas práticas de documentação e organização de código, incluindo modularização, comentários explicativos e estruturação lógica de notebooks para facilitar colaboração e manutenção.

Abertura e Contexto



A verdadeira utilidade dos dados está na capacidade de **resumir a realidade em números** que orientam decisões estratégicas.

▼ De: Filtradores de Dados

Profissionais que apenas consultam e filtram informações existentes, sem criar novos insights ou métricas de valor.



↗ Para: Analistas Estratégicos

Profissionais capazes de criar indicadores que direcionam decisões de negócio e medem o sucesso organizacional.

🎯 O Impacto dos Indicadores

Nesta aula, você aprenderá a transformar dados brutos em **KPIs acionáveis** que respondem perguntas críticas do negócio: Qual região tem melhor desempenho? Qual produto gera mais receita? Como está nossa base de clientes?

Agrupamento e Agregação

O Comando groupby()

O comando **groupby()** é a base para a criação de qualquer indicador em Pandas. Ele divide o DataFrame em grupos com base em uma ou mais colunas, permitindo aplicar funções de resumo a cada grupo separadamente.

```
# Sintaxe básica do groupby()  
df.groupby('Coluna')['Valor'].funcao()
```

O groupby() funciona em três etapas: **dividir** os dados em grupos, **aplicar** uma função a cada grupo e **combinar** os resultados em uma nova estrutura.

Conceito Fundamental

Imagine que você tem vendas de diferentes regiões. O groupby('Regiao') agrupa todas as vendas da mesma região, permitindo calcular totais, médias ou contagens específicas para cada região.

Funções de Agregação

As funções de agregação transformam os grupos criados pelo groupby() em um único valor resumido. Cada função serve a um propósito específico de análise.

Função	Descrição	Exemplo de Uso
mean()	Calcula a média	Ticket médio
sum()	Soma todos os valores	Receita total
count()	Conta o número de itens	Quantidade de vendas
max()	Retorna o valor máximo	Maior venda
min()	Retorna o valor mínimo	Menor venda

```
# Exemplo: Média de vendas por região  
df.groupby('Regiao')['Vendas'].mean()
```

```
# Exemplo: Total de vendas por produto  
df.groupby('Produto')['Vendas'].sum()
```

O Conceito de KPIs

🎯 O que é um KPI?

Um KPI é um **valor resumido** que ajuda a medir o sucesso de uma organização, departamento ou atividade específica em relação aos seus objetivos estratégicos.

KPI = Key Performance Indicator
(Indicador Chave de Desempenho)

⚙️ Como KPIs são Criados?

A maioria dos KPIs é criada por meio de **agrupamentos** e **agregações** de dados. Utilizamos o comando **groupby()** para dividir os dados em grupos e aplicamos funções de agregação para resumir cada grupo em um único valor significativo.

📌 Exemplos Práticos de KPIs

1. Ticket Médio

Valor médio gasto por transação ou cliente. Indica o poder de compra e ajuda a identificar oportunidades de upselling.

$$\text{Ticket Médio} = \text{Receita Total} / \text{Número de Transações}$$

2. Taxa de Conversão

Percentual de visitantes ou leads que se tornam clientes. Mede a eficácia do funil de vendas.

$$\text{Taxa de Conversão} = (\text{Clientes} / \text{Visitantes}) \times 100$$

3. Volume de Vendas por Período

Quantidade total de produtos vendidos em um período específico. Permite análise de tendências e sazonalidade.

$$\text{Volume} = \text{Soma de Quantidade por Mês/Trimestre/Ano}$$

Demonstração: Agrupamento e Agregação Avançada

Atividade Prática 1: Criação de Indicadores Sintéticos

Trabalhe em **grupos** para calcular três KPIs fundamentais usando o dataset de vendas fornecido. Registre todos os scripts e tabelas geradas.



KPI 1: Ticket Médio

Tarefa:

Calcular a média de valor por transação em cada região.

Cálculo:

```
df.groupby('Regiao')\n  ['Valor_Venda'].mean()
```

O que significa: Este KPI mostra quanto, em média, cada cliente gasta por compra em cada região. Valores mais altos indicam clientes com maior poder aquisitivo.



KPI 2: Volume de Vendas

Tarefa:

Calcular o total de vendas (quantidade) por produto.

Cálculo:

```
df.groupby('Produto')['Quantidade'].sum()
```

O que significa: Este KPI identifica quais produtos têm maior volume de vendas. Útil para gestão de estoque e estratégias de marketing.



KPI 3: Proporção de Clientes

Tarefa:

Contar o número de clientes em cada faixa etária e calcular a proporção.

Cálculo:

```
# Contagem
df['Faixa_Etaria'].value_counts()

# Proporção
df['Faixa_Etaria'].value_counts() /\nlen(df)
```

O que significa: Este KPI mostra a distribuição etária da base de clientes. Essencial para segmentação e personalização de campanhas.

Atividade Prática 2: Dashboard Sintético

Objetivo da Atividade

Consolidar os **três KPIs criados** na atividade anterior em um dashboard sintético simples, registrando os scripts e criando visualizações básicas para comunicar os resultados.

Passo a Passo Detalhado

1 Registro e Consolidação dos Scripts

Documente todos os scripts utilizados para calcular os três KPIs em um documento colaborativo (Google Docs ou Jamboard):

```
# KPI 1: Ticket Médio por Região
ticket_medio = df.groupby('Regiao')['Valor_Venda'].mean()

# KPI 2: Volume de Vendas por Produto
volume_vendas = df.groupby('Produto')['Quantidade'].sum()

# KPI 3: Proporção de Clientes por Faixa Etária
proporcao = df['Faixa_Etaria'].value_counts() / len(df)
```

2 Identificar os Indicadores

Para cada KPI, registre:

- **Nome do Indicador:** Ex: "Ticket Médio por Região"
- **Objetivo:** O que ele mede
- **Interpretação:** Como usar o resultado

3 Criar Visualizações Básicas

Use o Google Colab ou planilha para criar três gráficos simples:

```
# Gráfico de barras para Ticket Médio
ticket_medio.plot(kind='bar', title='Ticket Médio por Região')

# Gráfico de barras para Volume de Vendas
volume_vendas.plot(kind='bar', title='Volume de Vendas por Produto')

# Gráfico de pizza para Proporção de Clientes
proporcao.plot(kind='pie', title='Proporção de Clientes por Faixa Etária')
```

4 Montar o Dashboard Sintético

Organize os três gráficos em uma única visualização (pode ser no próprio notebook ou em uma planilha). Adicione títulos descritivos e legendas claras.

📋 Entregáveis

- Documento com scripts documentados
- Dashboard com três gráficos básicos
- Identificação clara de cada KPI
- Interpretação dos resultados

💡 Dica Profissional

Um dashboard eficaz conta uma história. Organize seus KPIs de forma que respondam perguntas sequenciais do negócio.

Filtragem Lógica para KPIs Específicos

▼ Operadores Lógicos

Use operadores lógicos **&** (E) e **|** (OU) para refinar os dados antes de calcular um KPI. Isso permite análises específicas para subconjuntos de dados.

Operador	Significado	Uso
&	E (AND)	Ambas condições devem ser verdadeiras
	OU (OR)	Pelo menos uma condição deve ser verdadeira
~	NÃO (NOT)	Inverte a condição

Exemplo 1: Filtro com E (&)

```
# KPI apenas para produtos Premium na região Sul
df[(df['Produto'] == 'Premium') &
(df['Regiao'] == 'Sul')].groupby('Regiao')['Vendas'].mean()
```

Exemplo 2: Filtro com OU (|)

```
# KPI para regiões Sul OU Norte
df[(df['Regiao'] == 'Sul') |
(df['Regiao'] == 'Norte')].groupby('Regiao')['Vendas'].sum()
```

📌 Filtro + KPI em Ação

Combine filtros com agrupamentos para criar KPIs altamente específicos que respondem perguntas de negócio precisas.

Exemplo 3: KPI Segmentado

```
# Ticket médio de clientes VIP por região
df[df['Cliente_VIP'] == True].groupby('Regiao')
['Valor_Venda'].mean()
```

Exemplo 4: Filtro Múltiplo

```
# Volume de vendas de produtos Básico
# com valor acima de 500 na região Norte
df[(df['Produto'] == 'Básico') &
(df['Valor_Venda'] > 500) &
(df['Regiao'] == 'Norte')]['Quantidade'].sum()
```






</> Boas Práticas de Scripting

- ✓ **Modularização:** Divida o código em células lógicas
- ✓ **Comentários:** Explique o porquê de cada groupby() ou filtro
- ✓ **Nomes descritivos:** Use variáveis com nomes claros
- ✓ **Parênteses:** Sempre use parênteses ao combinar filtros

Atividade Prática 3: Documentação de Scripts

Encerramento e Reflexão

✓ Principais Aprendizados

-  O comando **groupby()** é a base para criar qualquer indicador, dividindo dados em grupos para análise específica.
-  Funções de agregação (**mean, sum, count, max, min**) transformam grupos em valores únicos e significativos.
-  **KPIs acionáveis** são criados através de agrupamentos e agregações, medindo o sucesso do negócio.
-  Operadores lógicos (**& e |**) permitem refinar dados antes de calcular KPIs para análises específicas.
-  **Documentação e organização** de código são essenciais para validar e manter indicadores ao longo do tempo.

Reflexão Final

Um bom cientista de dados não apenas manipula os dados, mas também **cria as métricas que o negócio precisa** para medir o sucesso. A atenção aos detalhes e a organização são essenciais para a validade desses indicadores.

Próximos Passos

Continue praticando a criação de KPIs com diferentes datasets. Explore combinações de agrupamentos múltiplos e funções de agregação personalizadas. Lembre-se: **dados sem indicadores são apenas números**, mas indicadores bem construídos **direcionam decisões estratégicas**.