Aula 06: Outliers

Qual é a utilidade (ou o perigo) de ter outliers em nossa análise de dados?

Outlier é um ponto de dado que se desvia significativamente de outras observações. Em outras palavras, é um valor que está muito distante do restante dos dados.

1. **Importância:**
    - **Identificação de Erros:** Podem indicar erros de digitação, falhas de sensor ou medição.
    - **Insights de Negócios:** Podem revelar eventos raros, mas importantes (ex: um cliente que gasta muito mais que a média, um produto com defeito incomum, um pico de tráfego).
2. **Identificação:** Usamos métodos estatísticos e visualizações (como o Boxplot) para definir limites claros.
3. **Tratamento:** Depende do contexto:
    - Se for erro: Deve ser corrigido ou removido.
    - Se for evento real e importante: Deve ser analisado separadamente, mas pode ser removido das análises de média/regressão para evitar distorções.

Calculando e Identificando Outliers (IQR)

Um bom método para identificar outliers é através do **Intervalo Interquartil (IQR)**. Ele define um "cercadinho" ao redor dos 50% centrais dos dados. Vamos identificar os produtos que têm o preço muito acima ou muito abaixo do que é considerado normal no nosso conjunto de dados de exemplo (`vendas_produtos.csv`):

```
df_produtos = pd.read_csv('vendas_produtos.csv')

precos_array = df_produtos['preco'].values
```

A base dessa abordagem são os quartis Q1 e Q3:

1. **Quartis:** Marcam o ponto onde 25% e 75% dos dados estão, respectivamente.

2. **IQR:** É a amplitude dos 50% centrais dos dados:

   IQR = Q3 - Q1

3. **Limite Superior (LS):** Qualquer dado acima deste valor é considerado um **outlier superior**:

   LS = Q3 + (1.5 * IQR)

4. **Limite Inferior (LI):** Qualquer dado abaixo deste valor é considerado um **outlier inferior**.

   LI = Q1 - (1.5 * IQR)

Em Python, temos:

```
# Calcular Q1 e Q3
Q1 = np.percentile(precos_array, 25)
Q3 = np.percentile(precos_array, 75)

# Calcular IQR
IQR = Q3 - Q1

# Calcular Limites
limite_superior = Q3 + (1.5 * IQR)
limite_inferior = Q1 - (1.5 * IQR)

print(f"\n--- Limites de Outliers (Preços) ---")
print(f"Q1 (25%): R$ {Q1:.2f}")
print(f"Q3 (75%): R$ {Q3:.2f}")
print(f"IQR: R$ {IQR:.2f}")
print(f"Limite Superior (LS): R$ {limite_superior:.2f}")
print(f"Limite Inferior (LI): R$ {limite_inferior:.2f}")
```

OBS: A origem do '1.5' para o cálculo dos limites superior e inferior vem dos estudos de **John Tukey** na década de 1970, durante o desenvolvimento da **Análise Exploratória de Dados (EDA)** e, principalmente, do diagrama de caixa, o **Boxplot.**

Agora utilizamos **filtros e organizadores do Pandas** para encontrar os produtos (linhas) que ultrapassam esses limites:

```
# Identificação de Outliers Superiores e Inferiores
outliers_superiores = df_produtos[df_produtos['preco'] > limite_superior]
outliers_inferiores = df_produtos[df_produtos['preco'] < abs(limite_inferior)]

# Exibir Outliers Superiores Ordenados (Decrescente)
print(f"\n--- Outliers Superiores ({len(outliers_superiores)} produtos) ---")
print(outliers_superiores[['nome', 'preco']].sort_values(by='preco', ascending=False))

# Exibir Outliers Inferiores Ordenados (Crescente)
print(f"\n--- Outliers Inferiores ({len(outliers_inferiores)} produtos) ---")
print(outliers_inferiores[['nome', 'preco']].sort_values(by='preco', ascending=True))
```

Atividade Prática

1. **Partiremos da atividade anterior** (mesmo arquivo .csv).
2. **Verifiquem a presença de outliers** naquela variável quantitativa usando o método IQR.
3. Apresentem o resultado, dividindo os outliers em **superiores** e **inferiores**.

Durante nossa análise, podemos utilizar o matplotlib para criar um painel de visualização e demonstrar o que os outliers significam:

```
# Garante que os DataFrames de outliers não estão vazios antes de plotar
if not outliers_inferiores.empty or not outliers_superiores.empty:

    fig, axes = plt.subplots(1, 2, figsize=(14, 6)) # 1 linha, 2 colunas

    # 1ª Posição: Outliers Inferiores (Crescente)
    axes[0].bar(outliers_inferiores['nome'], outliers_inferiores['preco'])
    axes[0].set_title('Outliers Inferiores (Preços Mais Baixos)')
    axes[0].set_ylabel('Preço (R$)')
    axes[0].tick_params(axis='x', rotation=45, labelsize=8)
    axes[0].grid(axis='y', linestyle='--')

    # 2ª Posição: Outliers Superiores (Decrescente)
    # Ordenamos novamente para garantir a visualização correta
    outliers_superiores_plot = outliers_superiores.sort_values(by='preco', ascending=False)
    axes[1].bar(outliers_superiores_plot['nome'], outliers_superiores_plot['preco'])
    axes[1].set_title('Outliers Superiores (Preços Mais Altos)')
    axes[1].set_ylabel('Preço (R$)')
    axes[1].tick_params(axis='x', rotation=45, labelsize=8)
    axes[1].grid(axis='y', linestyle='--')

    plt.tight_layout() # Ajusta automaticamente os parâmetros de subplot para dar preenchimento
    plt.show()

else:
    print("\nNão houve outliers superiores ou inferiores para plotar.")
```