

Ciência de Dados (Python e Estatística)

```
# Adicionei os rótulos com os valores das vendas concluídas e pendentes no gráfico
for i in range(len(dados_vendedores)):
    plt.text(dados_vendedores['ID_vendedor'].iloc[i], dados_vendedores['valor_concluidas'].iloc[i],
             f"R${dados_vendedores['valor_concluidas'].iloc[i]:.2f}", fontsize=10, ha='right', va='bottom', color='black', weight='bold')
    plt.text(dados_vendedores['ID_vendedor'].iloc[i], dados_vendedores['valor_pendentes'].iloc[i],
             f"R${dados_vendedores['valor_pendentes'].iloc[i]:.2f}", fontsize=10, ha='right', va='bottom', color='black', weight='bold')

# Configurações do gráfico
plt.title('Comparação entre Metas de Vendas, Vendas Concluídas e Pendentes', fontsize=16, fontweight='bold')
plt.xlabel('ID Vendedor', fontsize=14)
plt.ylabel('Valor (R$)', fontsize=14)

# Melhorar a legibilidade dos números no eixo X e Y
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

# Adicionei legenda
plt.legend(loc='upper right', fontsize=12)

# Exibir o gráfico final
plt.tight_layout()
plt.show()
```

```
# Importei as bibliotecas necessárias
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Carreguei os dados dos arquivos CSV
vendedores = pd.read_csv("vendedores.csv")
vendas = pd.read_csv("vendas.csv")

# Separei as vendas concluídas e pendentes
vendas_concluidas = vendas[vendas['status'] == 'concluido']
vendas_pendentes = vendas[vendas['status'] == 'pendente']

# Somei as vendas realizadas (concluídas) por cada vendedor
vendas_concluidas_por_vendedor = vendas_concluidas.groupby('ID_vendedor')['valor'].sum().reset_index()
vendas_concluidas_por_vendedor.rename(columns={'valor': 'valor_concluidas'}, inplace=True)

# Somei as vendas pendentes por cada vendedor
vendas_pendentes_por_vendedor = vendas_pendentes.groupby('ID_vendedor')['valor'].sum().reset_index()
vendas_pendentes_por_vendedor.rename(columns={'valor': 'valor_pendentes'}, inplace=True)

# Mesclei as vendas concluídas e pendentes com os dados de vendedores
dados_vendedores = pd.merge(vendedores, vendas_concluidas_por_vendedor, on='ID_vendedor', how='left')
dados_vendedores = pd.merge(dados_vendedores, vendas_pendentes_por_vendedor, on='ID_vendedor', how='left')

# Preencher valores NaN com 0 (caso algum vendedor não tenha vendas concluídas ou pendentes)
dados_vendedores['valor_concluidas'].fillna(0, inplace=True)
dados_vendedores['valor_pendentes'].fillna(0, inplace=True)

# Configurei o estilo do gráfico
sns.set(style="whitegrid")

# Ajustei o tamanho do gráfico
plt.figure(figsize=(8, 6))

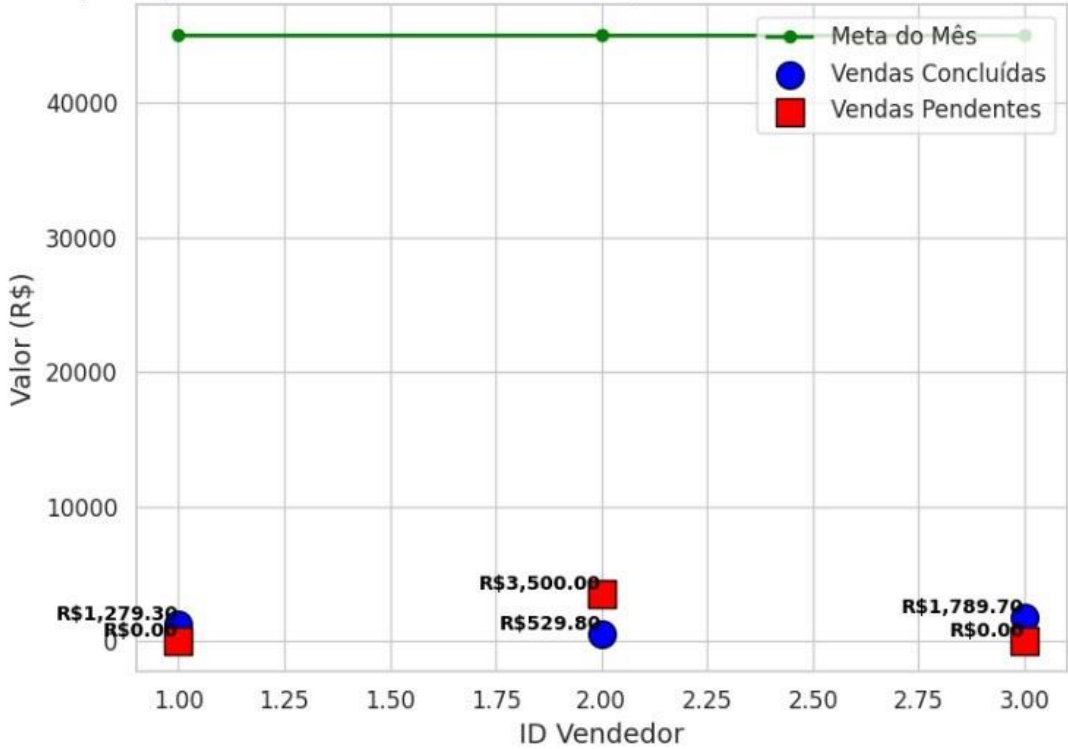
# Coloquei a linha da meta mensal
plt.plot(dados_vendedores['ID_vendedor'], dados_vendedores['meta_mes'], color='green', marker='o', label='Meta do Mês', linewidth=2)

# Coloquei os valores de vendas concluídas (bolinhas azuis)
plt.scatter(dados_vendedores['ID_vendedor'], dados_vendedores['valor_concluidas'], color='blue', s=200, label='Vendas Concluídas', marker='o', edgecolor='black')

# Coloquei os valores de vendas pendentes (quadrados vermelhos)
plt.scatter(dados_vendedores['ID_vendedor'], dados_vendedores['valor_pendentes'], color='red', s=200, label='Vendas Pendentes', marker='s', edgecolor='black')
```

Entrega 1: Análise Descritiva dos Dados:

Comparação entre Metas de Vendas, Vendas Concluídas e Pendentes



```
# Exibindo a matriz de confusão usando Seaborn para o Google Colab
plt.figure(figsize=(6, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Não Atingiu', 'Atingiu'], yticklabels=['Não Atingiu', 'Atingiu'])
plt.title('Matriz de Confusão - Desempenho de Vendedores')
plt.xlabel('Classificação Prevista')
plt.ylabel('Classificação Real')
plt.show()
```

```
# Importei as bibliotecas necessárias
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

# Carreguei os dados dos arquivos CSV (simulação)
vendedores = pd.read_csv("vendedores.csv")
vendas = pd.read_csv("vendas.csv")

# Separei as vendas concluídas e pendentes
vendas_concluidas = vendas[vendas['status'] == 'concluido']
vendas_pendentes = vendas[vendas['status'] == 'pendente']

# Somei as vendas realizadas (concluídas) por cada vendedor
vendas_concluidas_por_vendedor = vendas_concluidas.groupby('ID_vendedor')['valor'].sum().reset_index()
vendas_concluidas_por_vendedor.rename(columns={'valor': 'valor_concluidas'}, inplace=True)

# Somei as vendas pendentes por cada vendedor
vendas_pendentes_por_vendedor = vendas_pendentes.groupby('ID_vendedor')['valor'].sum().reset_index()
vendas_pendentes_por_vendedor.rename(columns={'valor': 'valor_pendentes'}, inplace=True)

# Mesclei as vendas concluídas e pendentes com os dados de vendedores
dados_vendedores = pd.merge(vendedores, vendas_concluidas_por_vendedor, on='ID_vendedor', how='left')
dados_vendedores = pd.merge(dados_vendedores, vendas_pendentes_por_vendedor, on='ID_vendedor', how='left')

# Preencher valores NaN com 0 (caso algum vendedor não tenha vendas concluídas ou pendentes)
dados_vendedores['valor_concluidas'].fillna(0, inplace=True)
dados_vendedores['valor_pendentes'].fillna(0, inplace=True)

# Definir a classificação real: se o vendedor atingiu ou não a meta
dados_vendedores['atingiu_meta_real'] = dados_vendedores['valor_concluidas'] >= dados_vendedores['meta_mes']

# Simulando uma previsão com base em um critério arbitrário, por exemplo:
# Classificamos como "atingiu" se o vendedor tem mais de R$5000,00 em vendas concluídas
dados_vendedores['atingiu_meta_prevista'] = dados_vendedores['valor_concluidas'] > 5000

# Gerando a matriz de confusão
conf_matrix = confusion_matrix(dados_vendedores['atingiu_meta_real'], dados_vendedores['atingiu_meta_prevista'])
```

Entrega 2: Modelagem Estatística:

Matriz de Confusão - Desempenho de Vendedores

