

```

import numpy as np
import pandas as pd
from faker import Faker
import warnings

# Ignorar FutureWarnings do pandas que podem aparecer com certas versões do
numpy/pandas
warnings.simplefilter(action='ignore', category=FutureWarning)

def gerar_pacientes_realistas(qtd=500,
file_path="pacientes_simulados_realistas_v3.csv"):
    """
    Gera um DataFrame e um arquivo CSV de pacientes diabéticos simulados (Versão 3).
    Os parâmetros são baseados na literatura fornecida sobre pé diabético,
    simulando dados de um sistema de palmilha inteligente (in-shoe).

    Atualizações v3 (baseadas nos novos artigos):
    - Adiciona 'retinopatia_s_n' e 'nefropatia_s_n' como fatores de risco.
    - Adiciona 'velocidade_marcha_m_s'.
    - Torna 'stance_time_s' (para PTI) dependente da velocidade da marcha.
      com DM.
    - Confirma faixas de Pressão (kPa) e Assimetria de Temperatura (°C).
    """

    faker = Faker('pt_BR')
    np.random.seed(42) # Para reprodutibilidade

    # --- PARÂMETROS DEMOGRÁFICOS E CLÍNICOS BASE ---
    idade_media = 58
    idade_std = 15
    # Tempo de diabetes > 10 anos é um fator de risco
    tempo_diabetes_media = 15
    imc_media = 30.0
    imc_std = 5.0

```

HbA1c > 9% é um fator de risco

hba1c_media = 8.8

hba1c_std = 1.8

--- PARÂMETROS CLÍNICOS E DE RISCO (Probabilidades) ---

p_neuropatia = 0.50 # Fator de risco primário

p_deformidade = 0.30 # Fator de risco

p_ulcera_previa = 0.25 # Fator de risco

p_amputacao_previa = 0.08 # Fator de risco

p_dap = 0.35 # Doença Arterial Periférica, fator de risco

p_retinopatia = 0.30 # Nova adição

p_nefropatia = 0.25 # Nova adição

p_has = 0.60

p_tabagismo = 0.25 # Fator de risco

p_alcool = 0.15

p_atividade_fisica = 0.40 # 1 = Ativo

--- PARÂMETROS FISIOLÓGICOS (SENSORES IN-SHOE) ---

1. PRESSÃO (Pico de Pressão Plantar - kPa)

A literatura (Reis et al., 2010) cita Boulton et al. (1983)

mencionando picos de > 10 kg/cm² (~980 kPa) em locais de úlcera.

Zhang et al. (2023) também mostra picos elevados nos metatarsos.

Simulamos valores "in-shoe", que são menores que "barefoot",

mas ainda elevados em pacientes de risco.

pressao_base_min_kpa = 80

pressao_base_max_kpa = 400

pressao_incremento_risco_kpa = 300 # Aumento para pacientes com
NP/Deformidade/Ulcera

pressao_std_dev_kpa = 100

2. TEMPERATURA (°C)

temp_media_normal_c = 29.0

temp_media_neuro_c = 32.0 # Neuropatia pode elevar a temp. base

```

temp_std_dev_c = 1.5
# Assimetria > 2.2°C é um indicador de risco crítico [cite: 27951, 33319]
temp_limiar_assimetria_c = 2.2
prob_assimetria_com_risco = 0.40 # Chance de simular "hot spot" em paciente de risco

# 3. UMIDADE (%)
humidity_range_perc = (30, 95) # Parâmetro comum em palmilhas

# 4. PARÂMETROS DA MARCHA
velocidade_media_m_s_base = 1.2
velocidade_std_m_s = 0.2
stance_time_media_s = 0.8 # Tempo de apoio médio
stance_time_std_s = 0.1 # Desvio padrão do tempo de apoio)

dados = []
for i in range(qtd):

    # --- PERFIL CLÍNICO DO PACIENTE ---
    idade = int(np.clip(np.random.normal(idade_media, idade_std), 25, 95))
    tempo_diabetes = int(np.clip(np.random.exponential(tempo_diabetes_media), 1,
60))
    imc = round(np.clip(np.random.normal(imc_media, imc_std), 18.5, 50), 1)
    hba1c = round(np.clip(np.random.normal(hba1c_media, hba1c_std), 5.0, 15.0), 1)

    # Fatores de Risco
    neuropatia = np.random.choice([0, 1], p=[1 - p_neuropatia, p_neuropatia])
    deformidade = np.random.choice([0, 1], p=[1 - p_deformidade, p_deformidade])
    dap = np.random.choice([0, 1], p=[1 - p_dap, p_dap])
    retinopatia = np.random.choice([0, 1], p=[1 - p_retinopatia, p_retinopatia])
    nefropatia = np.random.choice([0, 1], p=[1 - p_nefropatia, p_nefropatia])

    amputacao_previa = np.random.choice([0, 1], p=[1 - p_amputacao_previa,
p_amputacao_previa])

```

```

if amputacao_previa == 1:
    ulcera_previa = 1
else:
    ulcera_previa = np.random.choice([0, 1], p=[1 - p_ulcera_previa,
p_ulcera_previa])

has = np.random.choice([0, 1], p=[1 - p_has, p_has])
tabagismo = np.random.choice([0, 1], p=[1 - p_tabagismo, p_tabagismo])
alcool = np.random.choice([0, 1], p=[1 - p_alcool, p_alcool])
atividade_fisica = np.random.choice([0, 1], p=[1 - p_atividade_fisica,
p_atividade_fisica]) # 1 = Ativo

# --- LÓGICA DE RISCO DE ÚLCERA (CALCULADO) ---
# Baseado em Tavares et al. (2016)
pontos_risco = 0
if ulcera_previa == 1: pontos_risco += 5
if neuropatia == 1: pontos_risco += 3
if deformidade == 1: pontos_risco += 2
if amputacao_previa == 1: pontos_risco += 2
if dap == 1: pontos_risco += 1
if retinopatia == 1: pontos_risco += 1 # Novo
if nefropatia == 1: pontos_risco += 1 # Novo
if hba1c > 9.0: pontos_risco += 1
if tempo_diabetes > 20: pontos_risco += 1

risco_ulcera_calc = 1 if pontos_risco >= 5 else 0 # Alto Risco (ex: Neuro +
Deformidade)

# --- GERAÇÃO DE DADOS DOS SENSORES ---

# 1. Velocidade da Marcha (m/s)
# Reduz a velocidade média se houver alto risco (neuropatia/deformidade afeta
marcha)
velocidade_media_m_s = velocidade_media_m_s_base - (0.2 * risco_ulcera_calc)

```

```

    velocidade_marcha_m_s =
round(np.clip(np.random.normal(velocidade_media_m_s, velocidade_std_m_s), 0.5,
2.0), 2)

```

```

# 2. Pressão (Pico - kPa)
pressao_media = np.random.uniform(pressao_base_min_kpa,
pressao_base_max_kpa)
if risco_ulcera_calc == 1:
    pressao_media += np.random.uniform(50, pressao_incremento_risco_kpa)

```

```

# Pressão aumenta com a velocidade
pressao_media *= (1 + (velocidade_marcha_m_s - velocidade_media_m_s_base) *
0.5) # Fator de ajuste

```

```

p_esq = round(np.clip(np.random.normal(pressao_media, pressao_std_dev_kpa),
40, 1500), 2)
p_dir = round(np.clip(np.random.normal(pressao_media, pressao_std_dev_kpa *
1.1), 40, 1500), 2)

```

```

# 3. Pressão (Integral - PTI)
# Tempo de apoio (stance time) é inversamente proporcional à velocidade
# 3. Pressão (Integral - PTI)
# Tempo de apoio (stance time) é inversamente proporcional à velocidade
stance_time_esq = np.clip(np.random.normal(stance_time_media_s /
(velocidade_marcha_m_s / velocidade_media_m_s_base), stance_time_std_s), 0.5, 1.1)
stance_time_dir = np.clip(np.random.normal(stance_time_media_s /
(velocidade_marcha_m_s / velocidade_media_m_s_base), stance_time_std_s), 0.5, 1.1)#
PTI é a pressão acumulada ao longo do tempo de apoio
pti_esq = round(p_esq * stance_time_esq, 2)
pti_dir = round(p_dir * stance_time_dir, 2)

```

```

# 4. Temperatura (°C)
temp_media = temp_media_neuro_c if neuropatia == 1 else temp_media_normal_c

```

```
t_esq = round(np.clip(np.random.normal(temp_media, temp_std_dev_c), 20.0, 37.0), 1)
```

```
t_dir = round(np.clip(np.random.normal(temp_media, temp_std_dev_c), 20.0, 37.0), 1)
```

```
# Simula "Hot Spot" (Assimetria > 2.2°C) se houver alto risco
```

```
if risco_ulcera_calc == 1 and np.random.rand() < probab_assimetria_com_risco:
```

```
    diff = np.random.uniform(temp_limiar_assimetria_c, temp_limiar_assimetria_c + 2.5)
```

```
    if np.random.rand() < 0.5:
```

```
        t_dir = round(np.clip(t_esq + diff, 20.0, 38.5), 1) # Temp max. de inflamação
```

```
    else:
```

```
        t_esq = round(np.clip(t_dir + diff, 20.0, 38.5), 1)
```

```
temp_assimetria = round(abs(t_esq - t_dir), 1)
```

```
# 5. Umidade (%)
```

```
u_esq = round(np.random.uniform(*humidity_range_perc), 1)
```

```
u_dir = round(np.random.uniform(*humidity_range_perc), 1)
```

```
paciente = {
```

```
    # --- Perfil Clínico ---
```

```
    'id': f"PAC_{i+1:04d}",
```

```
    'nome': faker.first_name(),
```

```
    'sobrenome': faker.last_name(),
```

```
    'idade': idade,
```

```
    'sexo': np.random.choice(['M', 'F']),
```

```
    'tempo_diabetes_anos': tempo_diabetes,
```

```
    'hba1c_perc': hba1c,
```

```
    'imc': imc,
```

```
    'neuropatia_s_n': neuropatia,
```

```
    'deformidade_s_n': deformidade,
```

```
    'ulcera_previa_s_n': ulcera_previa,
```

```
    'amputacao_previa_s_n': amputacao_previa,
```

```
'dap_s_n': dap,
'retinopatia_s_n': retinopatia, # Novo
'nefropatia_s_n': nefropatia, # Novo
'has_s_n': has,
'tabagismo_s_n': tabagismo,
'alcool_s_n': alcool,
'atividade_fisica_s_n': atividade_fisica,
'risco_ulcera_calc': risco_ulcera_calc,
'velocidade_marcha_m_s': velocidade_marcha_m_s, # Novo
```

```
# --- Dados dos Sensores ---
```

```
'pressao_pico_esq_kpa': p_esq,
'pressao_pico_dir_kpa': p_dir,
'pressao_integral_esq_kpa_s': pti_esq,
'pressao_integral_dir_kpa_s': pti_dir,
'temperatura_esq_c': t_esq,
'temperatura_dir_c': t_dir,
'temp_assimetria_c': temp_assimetria,
'umidade_esq_perc': u_esq,
'umidade_dir_perc': u_dir
```

```
}
```

```
dados.append(paciente)
```

```
df = pd.DataFrame(dados)
```

```
# Reordenar colunas
```

```
colunas_perfil = [
    'id', 'nome', 'sobrenome', 'idade', 'sexo', 'tempo_diabetes_anos', 'hba1c_perc', 'imc',
    'neuropatia_s_n', 'deformidade_s_n', 'ulcera_previa_s_n', 'amputacao_previa_s_n',
    'dap_s_n', 'retinopatia_s_n', 'nefropatia_s_n', 'has_s_n',
    'tabagismo_s_n', 'alcool_s_n', 'atividade_fisica_s_n', 'risco_ulcera_calc',
    'velocidade_marcha_m_s'
```

```
]
```

```
colunas_sensores = [
```

```

        'pressao_pico_esq_kpa', 'pressao_pico_dir_kpa', 'pressao_integral_esq_kpa_s',
        'pressao_integral_dir_kpa_s',
        'temperatura_esq_c', 'temperatura_dir_c', 'temp_assimetria_c',
        'umidade_esq_perc', 'umidade_dir_perc'
    ]
    df = df[colunas_perfil + colunas_sensores]

```

Salvar em CSV com separador ; e decimal , (comum no Brasil)

try:

```

df.to_csv(file_path, index=False, sep=';', decimal=',')
print(f'Arquivo '{file_path}' gerado com {qtd} pacientes.")

```

Imprimir estatísticas de verificação

```
print("\n--- Verificação da Simulação (v3) ---")
```

if qtd > 0:

```
total_risco = df['risco_ulcera_calc'].sum()
```

```

print(f'Pacientes de Alto Risco ('risco_ulcera_calc' = 1): {total_risco} / {qtd}
({(total_risco/qtd)*100:.1f}%)")

```

if total_risco > 0:

```

media_assimetria_risco = df[df['risco_ulcera_calc'] ==
1]['temp_assimetria_c'].mean()

```

```

contagem_assimetria_risco = df[(df['temp_assimetria_c'] >
temp_limiar_assimetria_c) & (df['risco_ulcera_calc'] == 1)].shape[0]

```

```

print(f"      - Média da Assimetria de Temp. (Alto Risco):
{media_assimetria_risco:.2f}°C")

```

```

print(f"      - Pacientes de Alto Risco com Assimetria Crítica (>
{temp_limiar_assimetria_c}°C): {contagem_assimetria_risco}
({(contagem_assimetria_risco/total_risco)*100:.1f}%)")

```

```

media_pressao_risco = df[df['risco_ulcera_calc'] ==
1]['pressao_pico_esq_kpa'].mean()

```

```

media_pressao_normal = df[df['risco_ulcera_calc'] ==
0]['pressao_pico_esq_kpa'].mean()

```



```
print(f'Média Pico Pressão (Alto Risco): {media_pressao_risco:.2f} kPa")
print(f'Média Pico Pressão (Baixo Risco): {media_pressao_normal:.2f} kPa")
```

```
except Exception as e:
```

```
    print(f'Erro ao salvar o arquivo CSV: {e}')
```

```
return df
```

```
if __name__ == "__main__":
```

```
    gerar_pacientes_realistas(qtd=500,
```

```
    file_path="pacientes_simulados_v3_literatura.csv")
```