

Análise de Concentração e Desertos de Leitos por Macrorregião de Saúde

Identificação de Vazios Assistenciais e Desigualdades Regionais no Brasil

Cieges - Brasil Estadual

2026-01-28

Table of contents

1	Introdução	1
1.1	Objetivo	1
1.2	Metodologia	1
2	Análise Integrada por Macrorregião	2
2.1	Dados Gerais por Macrorregião	2
2.2	Análise de Concentração (HHI)	2
2.3	Visualização da Concentração	4
3	Classificação de Desertos Assistenciais	5
3.1	Critérios de Classificação	5
3.2	Mapa de Desertos por Macrorregião	6
4	Análise de Disponibilidade de UTI	7
4.1	Cobertura de Leitos Intensivos	7
4.2	Visualização da Cobertura UTI	8
5	Análise de Equidade e Desigualdade	9
5.1	Análise por Região Geográfica	9
5.2	Índice de Desigualdade	9
6	Indicadores Compostos e Ranking	10
6.1	Índice de Desempenho Assistencial	10
6.2	Visualização do Ranking	11
7	Recomendações e Conclusões	12
7.1	Macrorregiões Críticas	12
7.2	Oportunidades de Melhoria	13
7.3	Exportação de Resultados	13
8	Conclusões	14
8.1	Principais Achados	14
8.2	Recomendações Estratégicas	14

1 Introdução

1.1 Objetivo

Esta análise identifica **padrões de concentração de leitos hospitalares e desertos assistenciais** nas macrorregiões de saúde do Brasil, utilizando dados do CNES integrados com informações socioeconômicas do IBGE.

1.2 Metodologia

A análise combina indicadores de **disponibilidade, acessibilidade, concentração e desigualdade**:

Indicador	Fórmula	Interpretação
Leitos per capita	Leitos / 1.000 hab	OMS recomenda 3-5 leitos/1.000 hab
Índice HHI	$\Sigma(s^2)$	0-1 (maior = mais concentrado)
Cobertura UTI	% municípios com UTI	Acesso a cuidados intensivos
Razão SUS/Privado	Leitos SUS / Leitos totais	Equidade no acesso
IDH Médio	Média IDH dos municípios	Desenvolvimento humano

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

# Configuração visual
plt.rcParams['figure.dpi'] = 100
plt.rcParams['font.size'] = 10
sns.set_style("whitegrid")

# Carregar dados
print("Carregando bases de dados...")
df_cnes = pd.read_csv('arq2_tratado.csv', sep=';', encoding='latin1', low_memory=False)
df_ibge = pd.read_csv('dados macrorregião/TB_IBGE_DTB (1).csv', sep=';')

# Limpar dados IBGE
df_ibge['nu_area'] = df_ibge['nu_area'].astype(str).str.replace(',', '').astype(float)
df_ibge['nu_idh'] = df_ibge['nu_idh'].astype(str).str.replace(',', '.').astype(float)

print(f"Dados CNES: {len(df_cnes):,} registros")
print(f"Dados IBGE: {len(df_ibge):,} municípios")
print(f"Macrorregiões: {df_ibge['Macrorregião de Saúde'].nunique():,}")
```

Carregando bases de dados...
 Dados CNES: 49,804 registros
 Dados IBGE: 5,570 municípios
 Macrorregiões: 120

2 Análise Integrada por Macrorregião

2.1 Dados Gerais por Macrorregião

```
# Integrar dados CNES e IBGE
df_ibge['co_municipio_ibge'] = df_ibge['co_municipio_ibge'].astype(str)
df_cnes['codufmun'] = df_cnes['codufmun'].astype(str)

# Merge para enriquecer dados
df_merge = df_cnes.merge(
    df_ibge[['co_municipio_ibge', 'Macrorregião de Saúde', 'no_regiao', 'nu_idh', 'nu_area']],
    left_on='codufmun', right_on='co_municipio_ibge', how='left'
)

# Análise por macrorregião
macro_analise = df_merge.groupby('Macrorregião de Saúde').agg({
    'cnes': 'nunique',
    'codufmun': 'nunique',
    'qt_exist': 'sum',
    'qt_sus': 'sum',
    'qt_nsus': 'sum',
    'nu_idh': 'mean',
    'nu_area': 'sum'
}).round(2)

macro_analise.columns = ['Estabelecimentos', 'Municípios', 'Leitos Totais', 'Leitos SUS',
    'Leitos Não SUS', 'IDH Médio', 'Área (km²)']

# Calcular indicadores derivados
macro_analise['% SUS'] = (macro_analise['Leitos SUS'] / macro_analise['Leitos Totais'] * 100).round(1)
macro_analise['Leitos/Município'] = (macro_analise['Leitos Totais'] / macro_analise['Municípios']).round(1)
macro_analise['Leitos/100km²'] = (macro_analise['Leitos Totais'] / (macro_analise['Área (km²)'] / 100)).round(2)
macro_analise['Densidade Pop. (est)'] = (macro_analise['Municípios'] / (macro_analise['Área (km²)'] / 1000)).round(2)

macro_analise = macro_analise.sort_values('Leitos Totais', ascending=False)
macro_analise
```

Macrorregião de Saúde	Estabelecimentos	Municípios	Leitos Totais	Leitos SUS	Leitos Não SUS	IDH Médio	Área (km²)	% SUS	Leitos/Município	Leitos/100km²
RJ Macrorregiao I	654	87	44911	24187	20724	0.76	2.891021e+09	53.9	516.2	0.00
SP Rras6	342	1	38123	18238	19885	0.80	2.907380e+09	47.8	38123.0	0.00
PE Metropolitana	258	71	18581	11686	6895	0.72	2.408993e+08	62.9	261.7	0.01
MG Centro	219	50	17461	10176	7285	0.78	5.559649e+08	58.3	349.2	0.00
RS Metropolitana	113	45	14021	9450	4571	0.76	3.439050e+08	67.4	311.6	0.00
...
MS Costa Leste	24	11	913	630	283	0.72	5.044781e+08	69.0	83.0	0.00
AM Leste	19	16	884	847	37	0.61	2.484347e+09	95.8	55.2	0.00
MT Macrorregiao Oeste	17	12	762	632	130	0.70	9.018125e+08	82.9	63.5	0.00
MT Macrorregiao Leste	34	21	685	556	129	0.69	1.212429e+09	81.2	32.6	0.00
MS Pantanal	6	3	215	165	50	0.69	1.527999e+09	76.7	71.7	0.00

2.2 Análise de Concentração (HHI)

```
# Calcular Índice HHI por macrorregião
def calcular_hhi(df):
    """Calcular Índice Herfindahl-Hirschman"""
    total = df['qt_exist'].sum()
    if total == 0:
        return 1.0
    shares = df['qt_exist'] / total
    return (shares ** 2).sum()

hhi_macro = df_merge.groupby('Macrorregião de Saúde').apply(calcular_hhi).round(3)
hhi_macro = hhi_macro.sort_values(ascending=False)

# Classificação da concentração
def classificar_hhi(hhi):
    if hhi >= 0.25:
        return "ALTA CONCENTRAÇÃO"
    elif hhi >= 0.15:
        return "MÉDIA CONCENTRAÇÃO"
    else:
        return "BAIXA CONCENTRAÇÃO"

concentracao_class = hhi_macro.apply(classificar_hhi)

print("ÍNDICE DE CONCENTRAÇÃO HHI POR MACRORREGIÃO")
print("=" * 60)
for macro, hhi in hhi_macro.items():
    classe = concentracao_class[macro]
    print(f"{macro}: {hhi:.3f} - {classe}")
```

ÍNDICE DE CONCENTRAÇÃO HHI POR MACRORREGIÃO
 =====
 MS Pantanal: 0.080 - BAIXA CONCENTRAÇÃO
 SP Rras3: 0.060 - BAIXA CONCENTRAÇÃO

MG Vale Do Aco: 0.041 - BAIXA CONCENTRAÇÃO
 RR Macro-Roraima: 0.035 - BAIXA CONCENTRAÇÃO
 MT Macrorregiao Oeste: 0.024 - BAIXA CONCENTRAÇÃO
 SC Serra Catarinense: 0.023 - BAIXA CONCENTRAÇÃO
 SP Rras2: 0.023 - BAIXA CONCENTRAÇÃO
 SC Foz Do Rio Itajai: 0.022 - BAIXA CONCENTRAÇÃO
 AP Macro Unica - Ap: 0.022 - BAIXA CONCENTRAÇÃO
 MG Leste: 0.019 - BAIXA CONCENTRAÇÃO
 SP Rras4: 0.019 - BAIXA CONCENTRAÇÃO
 SP Rras11: 0.018 - BAIXA CONCENTRAÇÃO
 SP Rras 18: 0.017 - BAIXA CONCENTRAÇÃO
 MG Jequitinhonha: 0.016 - BAIXA CONCENTRAÇÃO
 MS Costa Leste: 0.016 - BAIXA CONCENTRAÇÃO
 RS Serra: 0.016 - BAIXA CONCENTRAÇÃO
 TO Macrorregiao De Saude Norte: 0.016 - BAIXA CONCENTRAÇÃO
 MG Noroeste: 0.016 - BAIXA CONCENTRAÇÃO
 AM Leste: 0.016 - BAIXA CONCENTRAÇÃO
 AL 2ª Macrorregiao De Saude: 0.016 - BAIXA CONCENTRAÇÃO
 CE Sertao Central: 0.015 - BAIXA CONCENTRAÇÃO
 MG Sudoeste: 0.015 - BAIXA CONCENTRAÇÃO
 MG Centro Sul: 0.015 - BAIXA CONCENTRAÇÃO
 PI Litoral: 0.015 - BAIXA CONCENTRAÇÃO
 CE Litoral Leste/Jaguaribe: 0.015 - BAIXA CONCENTRAÇÃO
 RS Sul: 0.014 - BAIXA CONCENTRAÇÃO
 SP Rras10: 0.014 - BAIXA CONCENTRAÇÃO
 ES Sul: 0.014 - BAIXA CONCENTRAÇÃO
 MG Triangulo Do Sul: 0.014 - BAIXA CONCENTRAÇÃO
 MT Macrorregiao Sul: 0.014 - BAIXA CONCENTRAÇÃO
 MG Leste Do Sul: 0.013 - BAIXA CONCENTRAÇÃO
 PI Cerrados: 0.013 - BAIXA CONCENTRAÇÃO
 MT Macrorregiao Centro-Noroeste: 0.013 - BAIXA CONCENTRAÇÃO
 MT Macrorregiao Leste: 0.013 - BAIXA CONCENTRAÇÃO
 SC Vale Do Itajai: 0.013 - BAIXA CONCENTRAÇÃO
 AM Oeste: 0.013 - BAIXA CONCENTRAÇÃO
 SC Sul: 0.012 - BAIXA CONCENTRAÇÃO
 SP Rras16: 0.012 - BAIXA CONCENTRAÇÃO
 SP Rras14: 0.012 - BAIXA CONCENTRAÇÃO
 PE Sertao: 0.012 - BAIXA CONCENTRAÇÃO
 SP Rras12: 0.012 - BAIXA CONCENTRAÇÃO
 AC Macro Unica - Ac: 0.012 - BAIXA CONCENTRAÇÃO
 BA Nordeste (Nrs - Alagoinhas): 0.012 - BAIXA CONCENTRAÇÃO
 GO Macrorregiao Nordeste: 0.011 - BAIXA CONCENTRAÇÃO
 SP Rras15: 0.011 - BAIXA CONCENTRAÇÃO
 BA Norte - (Nrs - Juazeiro): 0.011 - BAIXA CONCENTRAÇÃO
 BA Oeste (Nbs - Barreiras): 0.011 - BAIXA CONCENTRAÇÃO
 SC Grande Florianopolis: 0.011 - BAIXA CONCENTRAÇÃO
 RS Centro-Oeste: 0.011 - BAIXA CONCENTRAÇÃO
 PI Semi-Arido: 0.011 - BAIXA CONCENTRAÇÃO
 MT Macrorregiao Norte: 0.011 - BAIXA CONCENTRAÇÃO
 GO Macrorregiao Centro Sudeste: 0.011 - BAIXA CONCENTRAÇÃO
 MG Oeste: 0.010 - BAIXA CONCENTRAÇÃO
 SP Rras9: 0.010 - BAIXA CONCENTRAÇÃO
 SP Rras5: 0.010 - BAIXA CONCENTRAÇÃO
 SE Macro Unica: 0.010 - BAIXA CONCENTRAÇÃO
 SC Planalto Norte E Nordeste: 0.010 - BAIXA CONCENTRAÇÃO
 PI Meio Norte: 0.010 - BAIXA CONCENTRAÇÃO
 RO Macrorregiao I - Porto Velho: 0.010 - BAIXA CONCENTRAÇÃO
 GO Macrorregiao Sudoeste: 0.010 - BAIXA CONCENTRAÇÃO
 MG Nordeste: 0.010 - BAIXA CONCENTRAÇÃO
 MT Macrorregiao Centro-Norte: 0.010 - BAIXA CONCENTRAÇÃO
 ES Central Norte: 0.009 - BAIXA CONCENTRAÇÃO
 SC Meio Oeste: 0.009 - BAIXA CONCENTRAÇÃO
 PE Vale Do S.Francisco E Araripe: 0.009 - BAIXA CONCENTRAÇÃO
 RS Vales: 0.008 - BAIXA CONCENTRAÇÃO
 MS Centro: 0.008 - BAIXA CONCENTRAÇÃO
 PR Macrorregional Norte: 0.008 - BAIXA CONCENTRAÇÃO
 MS Cone Sul: 0.008 - BAIXA CONCENTRAÇÃO
 BA Extremo Sul (Nrs - Teixeira Freitas): 0.008 - BAIXA CONCENTRAÇÃO
 SP Rras17: 0.008 - BAIXA CONCENTRAÇÃO
 BA Centro - Norte (Nrs - Jacobina): 0.008 - BAIXA CONCENTRAÇÃO
 MG Sudeste: 0.008 - BAIXA CONCENTRAÇÃO
 SC Grande Oeste: 0.008 - BAIXA CONCENTRAÇÃO
 PB Macrorregiao Iii - Sertao/Alto Sertao: 0.008 - BAIXA CONCENTRAÇÃO
 PB Macrorregiao Ii - Campina Grande: 0.008 - BAIXA CONCENTRAÇÃO
 SP Rras8: 0.008 - BAIXA CONCENTRAÇÃO
 RN Macrorregiao Ii: 0.008 - BAIXA CONCENTRAÇÃO
 BA Sul (Nbs - Ilheus): 0.007 - BAIXA CONCENTRAÇÃO
 SP Rras1: 0.007 - BAIXA CONCENTRAÇÃO
 MA Macrorregiao Sul: 0.007 - BAIXA CONCENTRAÇÃO
 PA Macrorregional Ii: 0.007 - BAIXA CONCENTRAÇÃO
 MG Extremo Sul: 0.007 - BAIXA CONCENTRAÇÃO
 CE Sobral: 0.007 - BAIXA CONCENTRAÇÃO
 AM Central: 0.007 - BAIXA CONCENTRAÇÃO
 MG Sul: 0.007 - BAIXA CONCENTRAÇÃO
 MG Triangulo Do Norte: 0.007 - BAIXA CONCENTRAÇÃO
 SP Rras13: 0.006 - BAIXA CONCENTRAÇÃO
 SP Rras7: 0.006 - BAIXA CONCENTRAÇÃO
 MG Norte: 0.006 - BAIXA CONCENTRAÇÃO
 TO Macrorregiao Centro-Sul: 0.006 - BAIXA CONCENTRAÇÃO
 DF Distrito Federal: 0.006 - BAIXA CONCENTRAÇÃO
 AL 1ª Macrorregiao De Saude: 0.006 - BAIXA CONCENTRAÇÃO
 RS Metropolitana: 0.006 - BAIXA CONCENTRAÇÃO

GO Macrorregiao Centro-Norte: 0.006 - BAIXA CONCENTRAÇÃO
 PE Agreste: 0.006 - BAIXA CONCENTRAÇÃO
 RO Macrorregional Ii (Cacoal): 0.006 - BAIXA CONCENTRAÇÃO
 RN Macrorregiao I: 0.006 - BAIXA CONCENTRAÇÃO
 PA Macrorregional Iii: 0.006 - BAIXA CONCENTRAÇÃO
 RS Missioneira: 0.006 - BAIXA CONCENTRAÇÃO
 PB Macrorregiao I - Joao Pessoa: 0.006 - BAIXA CONCENTRAÇÃO
 CE Cariri: 0.005 - BAIXA CONCENTRAÇÃO
 BA Sudoeste (Nbs - Vitoria Conquista): 0.005 - BAIXA CONCENTRAÇÃO
 ES Metropolitana: 0.005 - BAIXA CONCENTRAÇÃO
 CE Fortaleza: 0.005 - BAIXA CONCENTRAÇÃO
 PR Macrorregional Noroeste: 0.005 - BAIXA CONCENTRAÇÃO
 RS Norte: 0.005 - BAIXA CONCENTRAÇÃO
 MA Macrorregiao Leste: 0.005 - BAIXA CONCENTRAÇÃO
 BA Centro-Leste (Nrs - Feira Santana): 0.005 - BAIXA CONCENTRAÇÃO
 MA Macrorregiao Norte: 0.004 - BAIXA CONCENTRAÇÃO
 PA Macrorregional Iv: 0.004 - BAIXA CONCENTRAÇÃO
 PR Macrorregional Leste: 0.004 - BAIXA CONCENTRAÇÃO
 PR Macrorregiao Oeste: 0.004 - BAIXA CONCENTRAÇÃO
 GO Macrorregiao Centro-Oeste: 0.004 - BAIXA CONCENTRAÇÃO
 BA Leste - (Nrs - Salvador): 0.003 - BAIXA CONCENTRAÇÃO
 PE Metropolitana: 0.003 - BAIXA CONCENTRAÇÃO
 MG Centro: 0.003 - BAIXA CONCENTRAÇÃO
 PA Macrorregional I: 0.003 - BAIXA CONCENTRAÇÃO
 SP Rras6: 0.002 - BAIXA CONCENTRAÇÃO
 RJ Macrorregiao I: 0.001 - BAIXA CONCENTRAÇÃO

2.3 Visualização da Concentração

```
fig, axes = plt.subplots(2, 2, figsize=(15, 10))

# 1. Top 15 Macrorregiões por Leitos
top15 = macro_analise.head(15)
bars1 = axes[0,0].barh(range(len(top15)), top15['Leitos Totais'], color='#2c3e50')
axes[0,0].set_yticks(range(len(top15)))
axes[0,0].set_yticklabels(top15.index, fontsize=8)
axes[0,0].set_xlabel('Leitos Totais')
axes[0,0].set_title('Top 15 Macrorregiões - Leitos Totais', fontweight='bold')
axes[0,0].grid(axis='x', alpha=0.3)

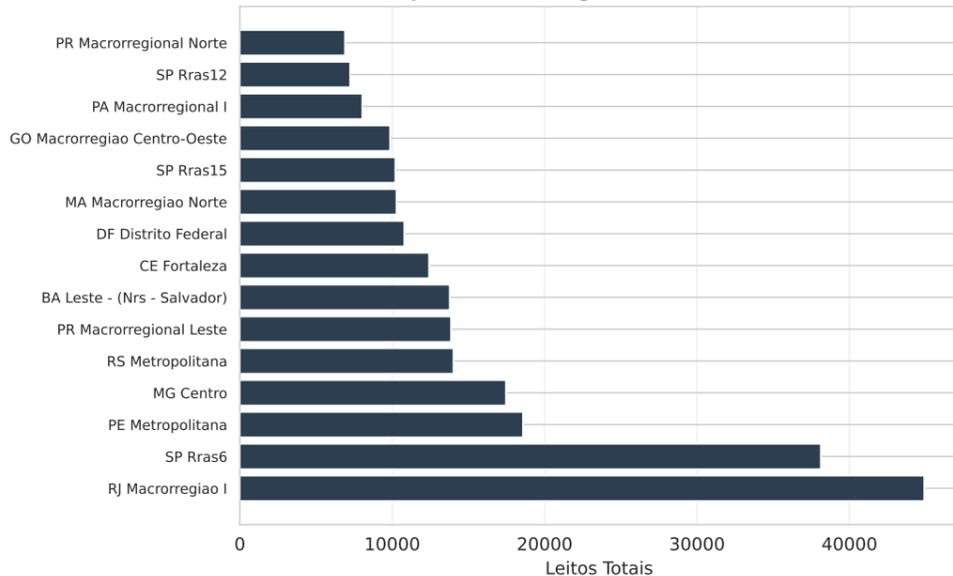
# 2. Índice HHI
hhi_top15 = hhi_macro.head(15)
bars2 = axes[0,1].barh(range(len(hhi_top15)), hhi_top15, color='#e74c3c')
axes[0,1].set_yticks(range(len(hhi_top15)))
axes[0,1].set_yticklabels(hhi_top15.index, fontsize=8)
axes[0,1].set_xlabel('Índice HHI')
axes[0,1].set_title('Top 15 - Índice de Concentração HHI', fontweight='bold')
axes[0,1].grid(axis='x', alpha=0.3)

# 3. Percentual SUS vs IDH
axes[1,0].scatter(macro_analise['IDH Médio'], macro_analise['% SUS'],
                  alpha=0.6, s=macro_analise['Leitos Totais']/100, c='#3498db')
axes[1,0].set_xlabel('IDH Médio')
axes[1,0].set_ylabel('% Leitos SUS')
axes[1,0].set_title('IDH vs % SUS (Tamanho = Leitos Totais)', fontweight='bold')
axes[1,0].grid(True, alpha=0.3)

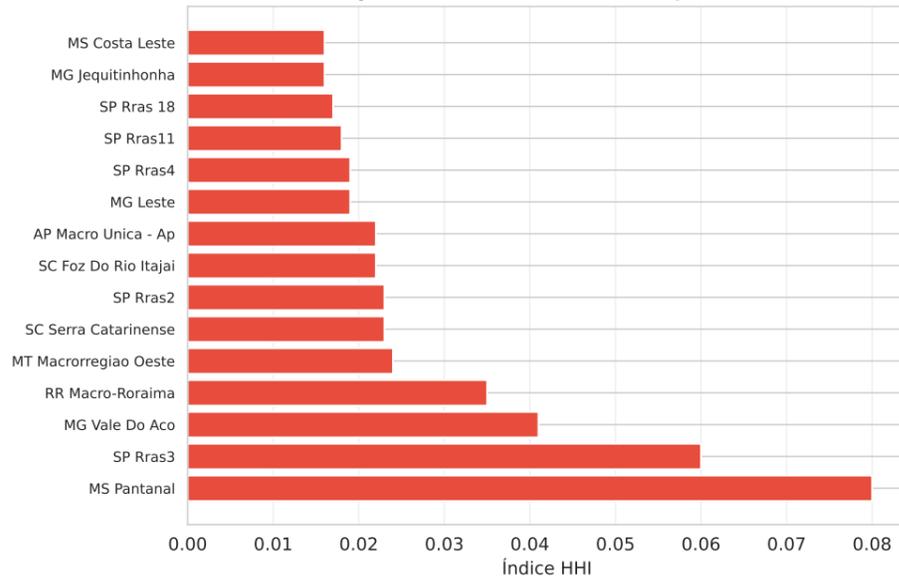
# 4. Leitos por Município vs Área
axes[1,1].scatter(macro_analise['Área (km²)'], macro_analise['Leitos/Município'],
                  alpha=0.6, s=macro_analise['IDH Médio']*100, c='#27ae60')
axes[1,1].set_xlabel('Área (km²)')
axes[1,1].set_ylabel('Leitos por Município')
axes[1,1].set_title('Área vs Leitos/Município (Tamanho = IDH)', fontweight='bold')
axes[1,1].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

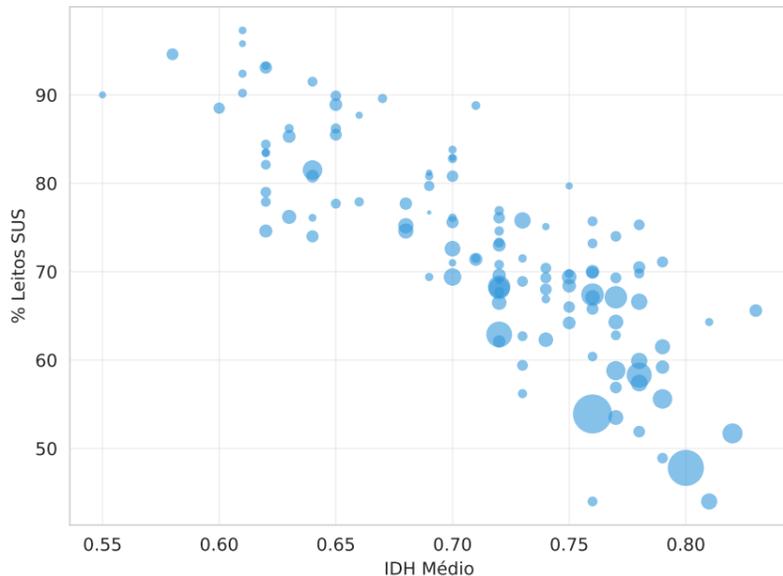
Top 15 Macrorregiões - Leitos Totais



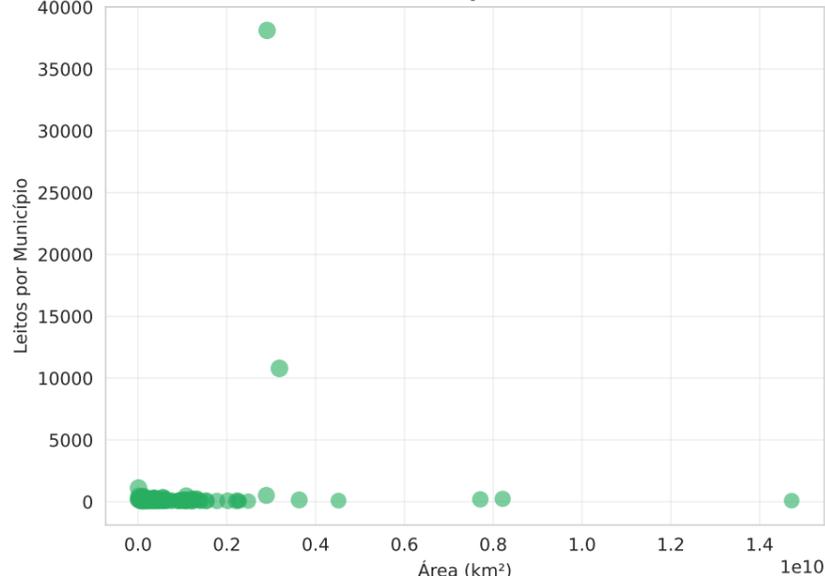
Top 15 - Índice de Concentração HHI



IDH vs % SUS (Tamanho = Leitos Totais)



Área vs Leitos/Município (Tamanho = IDH)



3 Classificação de Desertos Assistenciais

3.1 Critérios de Classificação

```
# Classificar municípios por disponibilidade de leitos
municipios_analise = df_merge.groupby(['Macrorregião de Saúde', 'codufmun']).agg({
    'qt_exist': 'sum',
    'qt_sus': 'sum',
    'cnes': 'nunique',
    'nu_idh': 'first'
}).reset_index()

# Critérios de classificação
def classificar_municipio(row):
    leitos = row['qt_exist']
    idh = row['nu_idh']

    if leitos == 0:
        return "DESERTO ABSOLUTO"
    elif leitos < 50:
        return "DESERTO CRÍTICO"
    elif leitos < 200:
        return "BAIXA DISPONIBILIDADE"
    elif leitos < 500:
        return "DISPONIBILIDADE MÉDIA"
    else:
        return "ALTA DISPONIBILIDADE"

municipios_analise['Classificacao'] = municipios_analise.apply(classificar_municipio, axis=1)

# Resumo por macrorregião
desertos_macro = municipios_analise.groupby(['Macrorregião de Saúde', 'Classificacao']).size().unstack(fill_value=0)
desertos_macro['Total'] = desertos_macro.sum(axis=1)
desertos_macro['% Desertos'] = (desertos_macro.get('DESERTO ABSOLUTO', 0) +
                                desertos_macro.get('DESERTO CRÍTICO', 0)) / desertos_macro['Total'] * 100

desertos_macro = desertos_macro.sort_values('% Desertos', ascending=False)
print("CLASSIFICAÇÃO DE DESERTOS ASSISTENCIAIS POR MACRORREGIÃO")
print("=" * 80)
print(desertos_macro.head(15))
```

CLASSIFICAÇÃO DE DESERTOS ASSISTENCIAIS POR MACRORREGIÃO

```
=====
Classificacao                ALTA DISPONIBILIDADE \
Macrorregião de Saúde
PI Semi-Arido                1
PI Cerrados                  0
PI Meio Norte                1
RN Macrorregiao Ii          1
```

RN Macrorregiao I	1
RR Macro-Roraima	1
GO Macrorregiao Centro-Oeste	2
MT Macrorregiao Leste	0
MS Centro	1
BA Centro - Norte (Nrs - Jacobina)	0
TO Macrorregiao Centro-Sul	1
AM Central	1
BA Oeste (Nbs - Barreiras)	1
PB Macrorregiao I - Joao Pessoa	1
PB Macrorregiao Iii - Sertao/Alto Sertao	0

Classificacao	BAIXA DISPONIBILIDADE \
Macrorregião de Saúde	
PI Semi-Arido	3
PI Cerrados	4
PI Meio Norte	2
RN Macrorregiao Ii	6
RN Macrorregiao I	10
RR Macro-Roraima	1
GO Macrorregiao Centro-Oeste	9
MT Macrorregiao Leste	4
MS Centro	4
BA Centro - Norte (Nrs - Jacobina)	6
TO Macrorregiao Centro-Sul	4
AM Central	4
BA Oeste (Nbs - Barreiras)	5
PB Macrorregiao I - Joao Pessoa	3
PB Macrorregiao Iii - Sertao/Alto Sertao	6

Classificacao	DESERTO CRÍTICO \
Macrorregião de Saúde	
PI Semi-Arido	32
PI Cerrados	37
PI Meio Norte	22
RN Macrorregiao Ii	51
RN Macrorregiao I	69
RR Macro-Roraima	10
GO Macrorregiao Centro-Oeste	49
MT Macrorregiao Leste	17
MS Centro	21
BA Centro - Norte (Nrs - Jacobina)	29
TO Macrorregiao Centro-Sul	24
AM Central	20
BA Oeste (Nbs - Barreiras)	27
PB Macrorregiao I - Joao Pessoa	23
PB Macrorregiao Iii - Sertao/Alto Sertao	34

Classificacao	DISPONIBILIDADE MÉDIA	Total \
Macrorregião de Saúde		
PI Semi-Arido	0	36
PI Cerrados	1	42
PI Meio Norte	0	25
RN Macrorregiao Ii	0	58
RN Macrorregiao I	1	81
RR Macro-Roraima	0	12
GO Macrorregiao Centro-Oeste	0	60
MT Macrorregiao Leste	0	21
MS Centro	0	26
BA Centro - Norte (Nrs - Jacobina)	1	36
TO Macrorregiao Centro-Sul	1	30
AM Central	0	25
BA Oeste (Nbs - Barreiras)	1	34
PB Macrorregiao I - Joao Pessoa	2	29
PB Macrorregiao Iii - Sertao/Alto Sertao	3	43

Classificacao	% Desertos
Macrorregião de Saúde	
PI Semi-Arido	88.888889
PI Cerrados	88.095238
PI Meio Norte	88.000000
RN Macrorregiao Ii	87.931034
RN Macrorregiao I	85.185185
RR Macro-Roraima	83.333333
GO Macrorregiao Centro-Oeste	81.666667
MT Macrorregiao Leste	80.952381
MS Centro	80.769231
BA Centro - Norte (Nrs - Jacobina)	80.555556
TO Macrorregiao Centro-Sul	80.000000
AM Central	80.000000
BA Oeste (Nbs - Barreiras)	79.411765
PB Macrorregiao I - Joao Pessoa	79.310345
PB Macrorregiao Iii - Sertao/Alto Sertao	79.069767

3.2 Mapa de Desertos por Macrorregião

```
# Análise detalhada dos desertos
desertos_detalhe = municipios_analise[
    municipios_analise['Classificacao'].isin(['DESERTO ABSOLUTO', 'DESERTO CRÍTICO'])
].groupby('Macrorregião de Saúde').agg({
    'codufmun': 'count',
    'qt_exist': ['sum', 'mean'],
```

```

    'nu_idh': 'mean'
}).round(2)

desertos_detalhe.columns = ['Nº Municípios', 'Total Leitos', 'Média Leitos', 'IDH Médio']
desertos_detalhe = desertos_detalhe.sort_values('Nº Municípios', ascending=False)

print("DETALHAMENTO DOS DESERTOS ASSISTENCIAIS")
print("=" * 80)
print(desertos_detalhe.head(10))

```

DETALHAMENTO DOS DESERTOS ASSISTENCIAIS

```

=====

```

	Nº Municípios	Total Leitos \
Macrorregião de Saúde		
MA Macrorregiao Norte	76	2002
RN Macrorregiao I	69	1015
RN Macrorregiao Ii	51	828
PR Macrorregional Norte	51	1116
GO Macrorregiao Centro-Oeste	49	936
PE Metropolitana	48	981
BA Centro-Leste (Nrs - Feira Santana)	48	1026
PR Macrorregional Noroeste	46	912
BA Sudoeste (Nbs - Vitoria Conquista)	45	1091
MA Macrorregiao Leste	41	1004

	Média Leitos	IDH Médio
Macrorregião de Saúde		
MA Macrorregiao Norte	26.34	0.57
RN Macrorregiao I	14.71	0.60
RN Macrorregiao Ii	16.24	0.61
PR Macrorregional Norte	21.88	0.70
GO Macrorregiao Centro-Oeste	19.10	0.70
PE Metropolitana	20.44	0.59
BA Centro-Leste (Nrs - Feira Santana)	21.38	0.58
PR Macrorregional Noroeste	19.83	0.70
BA Sudoeste (Nbs - Vitoria Conquista)	24.24	0.58
MA Macrorregiao Leste	24.49	0.56

4 Análise de Disponibilidade de UTI

4.1 Cobertura de Leitos Intensivos

```

# Identificar leitos UTI
def identificar_uti(row):
    """Classificar leitos como UTI ou não"""
    descricao = row['DS_CO_LEITO'].upper() if pd.notna(row['DS_CO_LEITO']) else ''
    codigo = row['co_leito']

    # Critérios para UTI
    uti_keywords = ['UTI', 'TERAPIA INTENSIVA', 'UNIDADE DE TERAPIA']
    uti_codigos = [74,75,76,77,78,79,80,81,82,83,85,86] # Códigos CNES UTI

    if any(keyword in descricao for keyword in uti_keywords) or codigo in uti_codigos:
        return 'UTI'
    else:
        return 'NÃO_UTI'

df_merge['Tipo_Leito'] = df_merge.apply(identificar_uti, axis=1)

# Análise de UTI por macrorregião
uti_macro = df_merge.groupby(['Macrorregião de Saúde', 'Tipo_Leito']).agg({
    'qt_exist': 'sum',
    'codufmun': 'nunique'
}).unstack(fill_value=0)

# Verificar estrutura das colunas
uti_macro.columns = [f'{col[0]}_{col[1]}' if col[1] else col[0] for col in uti_macro.columns]

# Renomear colunas
col_mapping = {}
for col in uti_macro.columns:
    if 'qt_exist_NÃO_UTI' in col:
        col_mapping[col] = 'Leitos Não UTI'
    elif 'qt_exist_UTI' in col:
        col_mapping[col] = 'Leitos UTI'
    elif 'codufmun_NÃO_UTI' in col:
        col_mapping[col] = 'Municípios Não UTI'
    elif 'codufmun_UTI' in col:
        col_mapping[col] = 'Municípios UTI'

uti_macro = uti_macro.rename(columns=col_mapping)
uti_macro['Total Leitos'] = uti_macro.get('Leitos Não UTI', 0) + uti_macro.get('Leitos UTI', 0)
uti_macro['% UTI'] = (uti_macro['Leitos UTI'] / uti_macro['Total Leitos'] * 100).round(1)
uti_macro['Municípios c/ UTI'] = (df_merge[df_merge['Tipo_Leito'] == 'UTI']
    .groupby('Macrorregião de Saúde')['codufmun'].nunique())
uti_macro['Total Municípios'] = df_merge.groupby('Macrorregião de Saúde')['codufmun'].nunique()
uti_macro['% Municípios c/ UTI'] = (uti_macro['Municípios c/ UTI'] /
    uti_macro['Total Municípios'] * 100).round(1)

uti_macro = uti_macro.sort_values('% UTI', ascending=False)
print("ANÁLISE DE DISPONIBILIDADE DE UTI POR MACRORREGIÃO")
print("=" * 80)

```

```
print(uti_macro.head(15))
```

ANÁLISE DE DISPONIBILIDADE DE UTI POR MACRORREGIÃO

Macrorregião de Saúde	Leitos Não UTI	Leitos UTI \
MT Macrorregiao Centro-Norte	2558	799
SP Rras6	30195	7928
DF Distrito Federal	8554	2234
RJ Macrorregiao I	35598	9313
SP Rras1	5312	1380
SP Rras7	3484	902
BA Leste - (Nrs - Salvador)	10973	2801
MG Vale Do Aco	1229	283
MG Centro	14219	3242
ES Metropolitana	5587	1256
RO Macrorregiao I - Porto Velho	2895	598
SP Rras16	2088	423
SP Rras5	2288	454
MG Triangulo Do Norte	2848	566
MT Macrorregiao Norte	1643	326

Macrorregião de Saúde	Municípios Não UTI	Municípios UTI \
MT Macrorregiao Centro-Norte	7	2
SP Rras6	1	1
DF Distrito Federal	1	1
RJ Macrorregiao I	87	54
SP Rras1	6	6
SP Rras7	13	8
BA Leste - (Nrs - Salvador)	42	9
MG Vale Do Aco	9	4
MG Centro	50	19
ES Metropolitana	20	8
RO Macrorregiao I - Porto Velho	18	6
SP Rras16	14	7
SP Rras5	6	6
MG Triangulo Do Norte	20	7
MT Macrorregiao Norte	24	7

Macrorregião de Saúde	Total Leitos	% UTI	Municípios c/ UTI \
MT Macrorregiao Centro-Norte	3357	23.8	2
SP Rras6	38123	20.8	1
DF Distrito Federal	10788	20.7	1
RJ Macrorregiao I	44911	20.7	54
SP Rras1	6692	20.6	6
SP Rras7	4386	20.6	8
BA Leste - (Nrs - Salvador)	13774	20.3	9
MG Vale Do Aco	1512	18.7	4
MG Centro	17461	18.6	19
ES Metropolitana	6843	18.4	8
RO Macrorregiao I - Porto Velho	3493	17.1	6
SP Rras16	2511	16.8	7
SP Rras5	2742	16.6	6
MG Triangulo Do Norte	3414	16.6	7
MT Macrorregiao Norte	1969	16.6	7

Macrorregião de Saúde	Total Municípios	% Municípios c/ UTI
MT Macrorregiao Centro-Norte	7	28.6
SP Rras6	1	100.0
DF Distrito Federal	1	100.0
RJ Macrorregiao I	87	62.1
SP Rras1	6	100.0
SP Rras7	13	61.5
BA Leste - (Nrs - Salvador)	42	21.4
MG Vale Do Aco	9	44.4
MG Centro	50	38.0
ES Metropolitana	20	40.0
RO Macrorregiao I - Porto Velho	18	33.3
SP Rras16	14	50.0
SP Rras5	6	100.0
MG Triangulo Do Norte	20	35.0
MT Macrorregiao Norte	24	29.2

4.2 Visualização da Cobertura UTI

```
fig, axes = plt.subplots(1, 2, figsize=(15, 6))

# 1. Percentual de Leitos UTI
uti_top15 = uti_macro.head(15)
bars1 = axes[0].barh(range(len(uti_top15)), uti_top15['% UTI'], color='#9b59b6')
axes[0].set_yticks(range(len(uti_top15)))
axes[0].set_yticklabels(uti_top15.index, fontsize=8)
axes[0].set_xlabel('% Leitos UTI')
axes[0].set_title('Top 15 - % Leitos UTI', fontweight='bold')
axes[0].grid(axis='x', alpha=0.3)

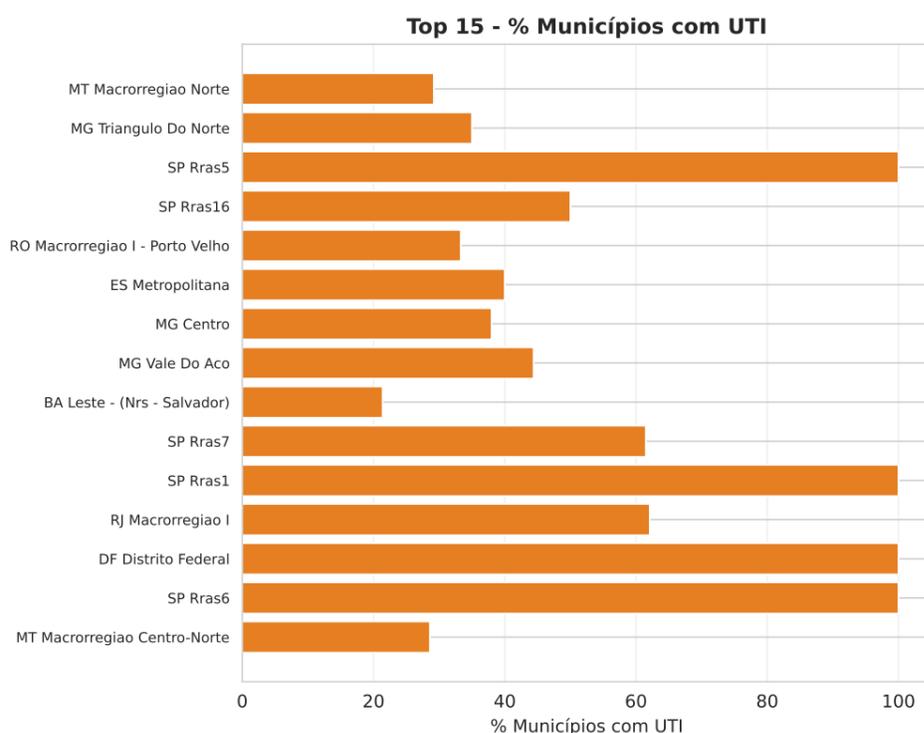
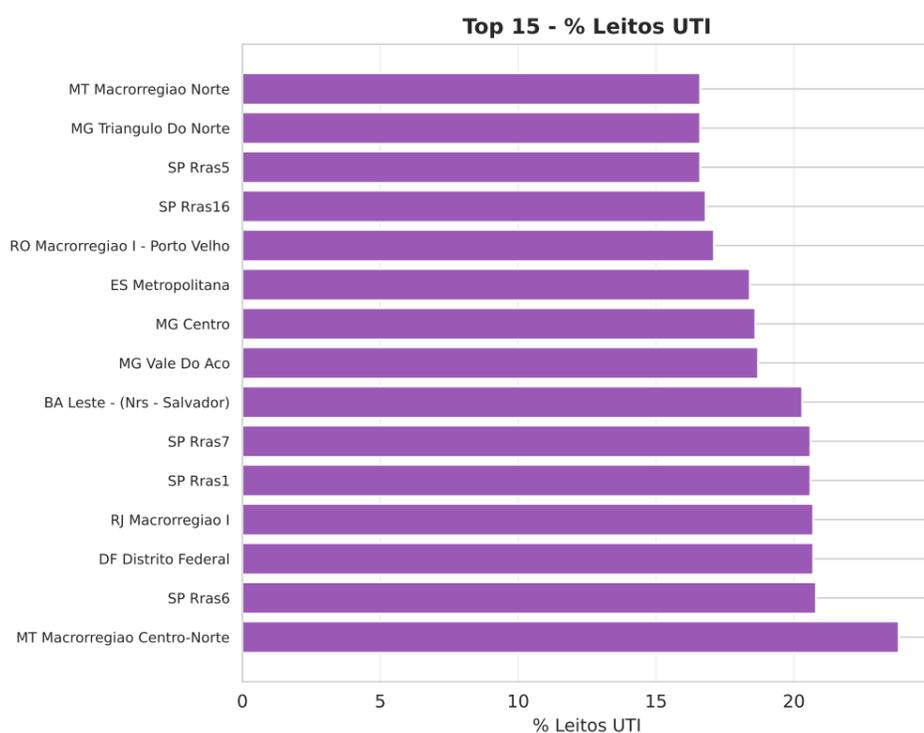
# 2. Percentual de Municípios com UTI
bars2 = axes[1].barh(range(len(uti_top15)), uti_top15['% Municípios c/ UTI'], color='#e67e22')
axes[1].set_yticks(range(len(uti_top15)))
```

```

axes[1].set_yticklabels(uti_top15.index, fontsize=8)
axes[1].set_xlabel('% Municípios com UTI')
axes[1].set_title('Top 15 - % Municípios com UTI', fontweight='bold')
axes[1].grid(axis='x', alpha=0.3)

plt.tight_layout()
plt.show()

```



5 Análise de Equidade e Desigualdade

5.1 Análise por Região Geográfica

```

# Agrupar por grande região geográfica
regiao_geo = df_merge.groupby('no_regiao').agg({
    'cnes': 'nunique',
    'codufmun': 'nunique',
    'qt_exist': 'sum',
    'qt_sus': 'sum',
    'nu_idh': 'mean',
    'nu_area': 'sum'
}).round(2)

regiao_geo.columns = ['Estabelecimentos', 'Municípios', 'Leitos Totais', 'Leitos SUS',
    'IDH Médio', 'Área (km²)']

regiao_geo['% SUS'] = (regiao_geo['Leitos SUS'] / regiao_geo['Leitos Totais'] * 100).round(1)
regiao_geo['Leitos/Município'] = (regiao_geo['Leitos Totais'] / regiao_geo['Municípios']).round(1)
regiao_geo['Leitos/100km²'] = (regiao_geo['Leitos Totais'] / (regiao_geo['Área (km²)'] / 100)).round(2)

regiao_geo = regiao_geo.sort_values('Leitos Totais', ascending=False)
print("ANÁLISE POR REGIÃO GEOGRÁFICA")
print("=" * 60)
print(regiao_geo)

```

ANÁLISE POR REGIÃO GEOGRÁFICA

```

=====
no_regiao      Estabelecimentos  Municípios  Leitos Totais  Leitos SUS  \
Regiao Sudeste          3132           914          221530      128125
Regiao Nordeste          2955          1355          142405      108138
Regiao Sul               1217           634           83637       57834
Regiao Centro-Oeste       973           361           47677       30102
Regiao norte              795           333           39884       30980

no_regiao      IDH Médio  Área (km²)  % SUS  Leitos/Município  \
Regiao Sudeste    0.76  1.559629e+10  57.8      242.4
Regiao Nordeste    0.67  1.414798e+10  75.9      105.1
Regiao Sul         0.76  6.354255e+09  69.1      131.9
Regiao Centro-Oeste 0.74  2.290175e+10  63.1      132.1
Regiao norte       0.67  5.029953e+10  77.7      119.8

no_regiao      Leitos/100km²
Regiao Sudeste    0.0
Regiao Nordeste    0.0
Regiao Sul         0.0
Regiao Centro-Oeste 0.0
Regiao norte       0.0

```

5.2 Índice de Desigualdade

```

# Calcular coeficiente de Gini para distribuição de leitos
def calcular_gini(values):
    """Calcular coeficiente de Gini"""

```

```

values = sorted(values)
n = len(values)
if n == 0:
    return 0

cumulative = np.cumsum(values)
gini = (n + 1 - 2 * np.sum(cumulative) / cumulative[-1]) / n
return gini

# Gini por macrorregião
gini_macro = df_merge.groupby('Macrorregião de Saúde').apply(
    lambda x: calcular_gini(x['qt_exist'].values)
).round(3)

gini_macro = gini_macro.sort_values(ascending=False)

print("COEFICIENTE DE GINI - DESIGUALDADE NA DISTRIBUIÇÃO DE LEITOS")
print("=" * 70)
print("(0 = perfeita igualdade, 1 = máxima desigualdade)")
print()
for macro, gini in gini_macro.head(15).items():
    print(f"{macro}: {gini:.3f}")

```

```

COEFICIENTE DE GINI - DESIGUALDADE NA DISTRIBUIÇÃO DE LEITOS
=====
(0 = perfeita igualdade, 1 = máxima desigualdade)

RR Macro-Roraima: 0.681
RS Metropolitana: 0.671
MG Vale Do Aco: 0.659
SP Rras17: 0.657
SC Grande Florianopolis: 0.655
SP Rras9: 0.650
SP Rras12: 0.636
PR Macrorregional Leste: 0.634
SP Rras15: 0.630
MG Centro: 0.630
CE Fortaleza: 0.629
PR Macrorregional Norte: 0.627
GO Macrorregiao Centro-Oeste: 0.625
SP Rras10: 0.619
RO Macrorregiao I - Porto Velho: 0.615

```

6 Indicadores Compostos e Ranking

6.1 Índice de Desempenho Assistencial

```

# Criar índice composto
def criar_indice_desempenho(df):
    """Índice composto de desempenho assistencial"""

    # Normalizar indicadores (0-1)
    leitos_capita_norm = (df['Leitos/Município'] - df['Leitos/Município'].min()) / \
        (df['Leitos/Município'].max() - df['Leitos/Município'].min())

    sus_norm = df['% SUS'] / 100

    idh_norm = (df['IDH Médio'] - df['IDH Médio'].min()) / \
        (df['IDH Médio'].max() - df['IDH Médio'].min())

    # Índice de concentração (invertido - menor HHI = melhor)
    hhi_values = df_merge.groupby('Macrorregião de Saúde').apply(calcular_hhi)
    hhi_norm = 1 - (hhi_values - hhi_values.min()) / (hhi_values.max() - hhi_values.min())

    # Índice composto (pesos diferentes)
    indice = (leitos_capita_norm * 0.3 +
              sus_norm * 0.3 +
              idh_norm * 0.2 +
              hhi_norm * 0.2)

    return indice.round(3)

# Calcular índice de desempenho
macro_analise['Índice Desempenho'] = criar_indice_desempenho(macro_analise)
macro_analise['Ranking'] = macro_analise['Índice Desempenho'].rank(ascending=False, method='min')

# Apresentar ranking
ranking_final = macro_analise[['Índice Desempenho', 'Ranking', 'Leitos Totais', 'Leitos/Município',
                               '% SUS', 'IDH Médio']].sort_values('Ranking')

print("RANKING DE DESEMPENHO ASSISTENCIAL POR MACRORREGIÃO")
print("=" * 80)
print("(Índice composto: disponibilidade, equidade, desenvolvimento, concentração)")
print()
print(ranking_final.head(20))

```

```

RANKING DE DESEMPENHO ASSISTENCIAL POR MACRORREGIÃO
=====
(Índice composto: disponibilidade, equidade, desenvolvimento, concentração)

                Índice Desempenho  Ranking  Leitos Totais  \
Macrorregião de Saúde

```

SP Rras6	0.820	1.0	38123
DF Distrito Federal	0.621	2.0	10788
SC Grande Florianopolis	0.573	3.0	3856
SC Planalto Norte E Nordeste	0.564	4.0	2830
SC Vale Do Itajai	0.561	5.0	2689
SC Grande Oeste	0.560	6.0	2181
SC Sul	0.553	7.0	2558
SP Rras13	0.553	7.0	6866
PR Macrorregional Leste	0.552	9.0	13855
SC Meio Oeste	0.549	10.0	2095
MG Triangulo Do Norte	0.545	11.0	3414
PR Macrorregiao Oeste	0.545	11.0	5361
SP Rras5	0.544	13.0	2742
SP Rras10	0.542	14.0	3463
RS Metropolitana	0.542	14.0	14021
PR Macrorregional Noroeste	0.541	16.0	5243
SP Rras17	0.540	17.0	5811
PR Macrorregional Norte	0.539	18.0	6900
SP Rras7	0.538	19.0	4386
RS Norte	0.538	19.0	4526

	Leitos/Município	% SUS	IDH Médio
Macrorregião de Saúde			
SP Rras6	38123.0	47.8	0.80
DF Distrito Federal	10788.0	51.7	0.82
SC Grande Florianopolis	275.4	65.6	0.83
SC Planalto Norte E Nordeste	176.9	71.1	0.79
SC Vale Do Itajai	112.0	75.3	0.78
SC Grande Oeste	66.1	75.7	0.76
SC Sul	106.6	74.0	0.77
SP Rras13	176.1	66.6	0.78
PR Macrorregional Leste	230.9	67.1	0.77
SC Meio Oeste	77.6	73.2	0.76
MG Triangulo Do Norte	170.7	69.9	0.76
PR Macrorregiao Oeste	105.1	69.4	0.75
SP Rras5	457.0	69.3	0.77
SP Rras10	138.5	70.5	0.78
RS Metropolitana	311.6	67.4	0.76
PR Macrorregional Noroeste	88.9	67.1	0.76
SP Rras17	242.1	61.5	0.79
PR Macrorregional Norte	95.8	75.8	0.73
SP Rras7	337.4	59.2	0.79
RS Norte	87.0	68.4	0.75

6.2 Visualização do Ranking

```

fig, axes = plt.subplots(2, 2, figsize=(16, 12))

# 1. Top 15 - Índice de Desempenho
top15_desempenho = ranking_final.head(15)
bars1 = axes[0,0].barh(range(len(top15_desempenho)), top15_desempenho['Índice Desempenho'],
                        color='#16a085')
axes[0,0].set_yticks(range(len(top15_desempenho)))
axes[0,0].set_yticklabels(top15_desempenho.index, fontsize=8)
axes[0,0].set_xlabel('Índice de Desempenho')
axes[0,0].set_title('Top 15 - Índice de Desempenho Assistencial', fontweight='bold')
axes[0,0].grid(axis='x', alpha=0.3)

# 2. Dispersão: Desempenho vs Leitos Totais
axes[0,1].scatter(macro_analise['Leitos Totais'], macro_analise['Índice Desempenho'],
                  alpha=0.6, s=60, c=macro_analise['IDH Médio'], cmap='viridis')
axes[0,1].set_xlabel('Leitos Totais')
axes[0,1].set_ylabel('Índice de Desempenho')
axes[0,1].set_title('Desempenho vs Volume (Cor = IDH)', fontweight='bold')
axes[0,1].grid(True, alpha=0.3)

# 3. Box Plot do Índice por Região Geográfica
regioes_box = []
for regioao in macro_analise.index:
    if regioao in df_merge['Macrorregião de Saúde'].values:
        regioao_geo = df_merge[df_merge['Macrorregião de Saúde'] == regioao]['no_regiao'].iloc[0]
        regioes_box.append((regioao, regioao_geo, macro_analise.loc[regioao, 'Índice Desempenho']))

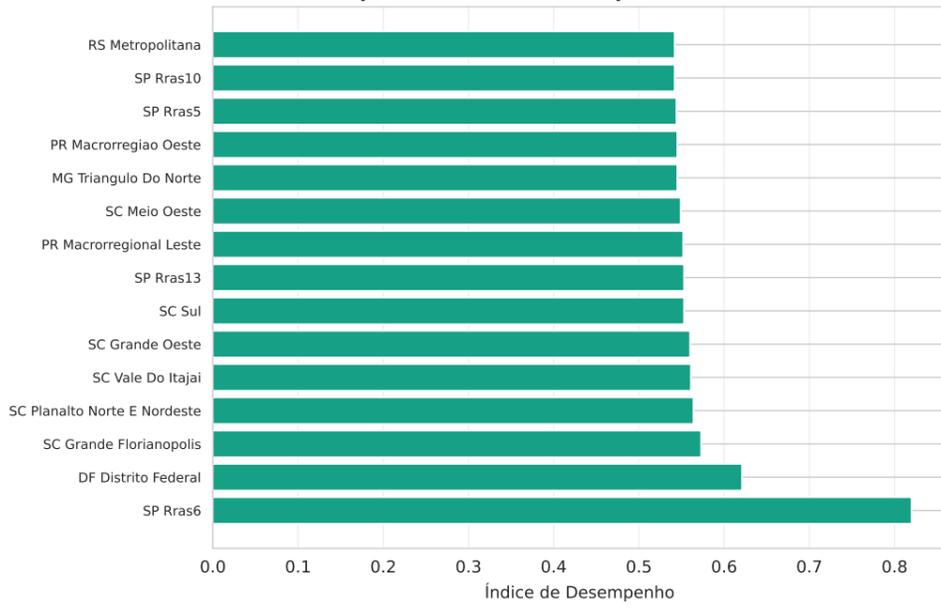
df_box = pd.DataFrame(regioes_box, columns=['Macrorregião', 'Região', 'Índice'])
sns.boxplot(data=df_box, x='Região', y='Índice', ax=axes[1,0])
axes[1,0].set_title('Distribuição do Índice por Região Geográfica', fontweight='bold')
axes[1,0].tick_params(axis='x', rotation=45)

# 4. Heatmap de Correlações
corr_data = macro_analise[['Leitos Totais', 'Leitos/Município', '% SUS', 'IDH Médio',
                           'Índice Desempenho']].corr()
sns.heatmap(corr_data, annot=True, cmap='coolwarm', center=0, ax=axes[1,1],
            fmt='.2f', cbar_kws={'shrink': 0.8})
axes[1,1].set_title('Matriz de Correlação', fontweight='bold')

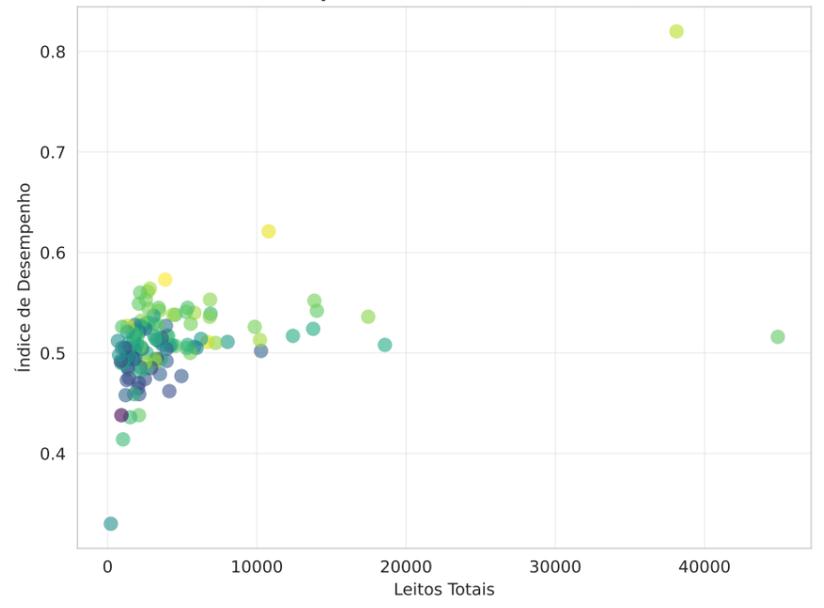
plt.tight_layout()
plt.show()

```

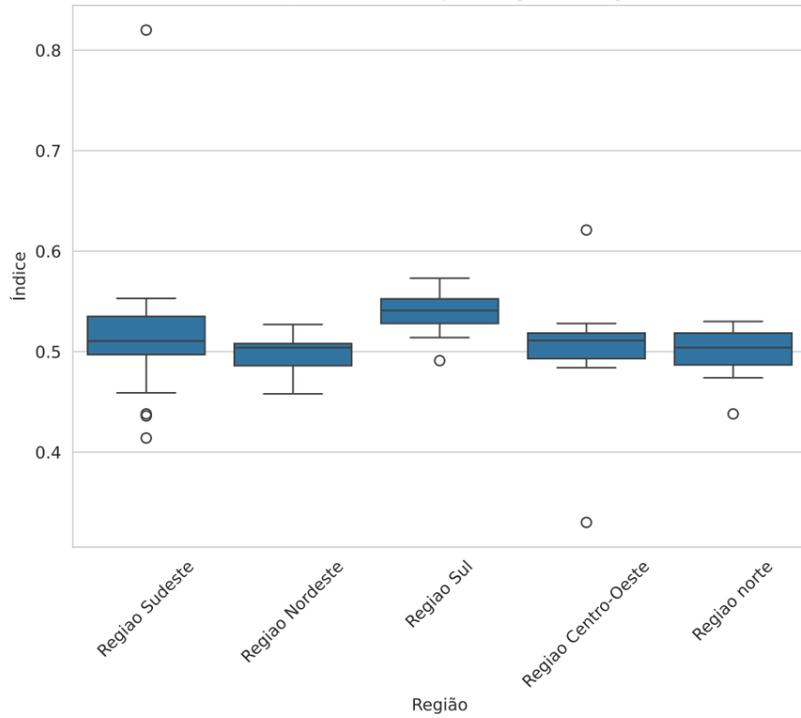
Top 15 - Índice de Desempenho Assistencial



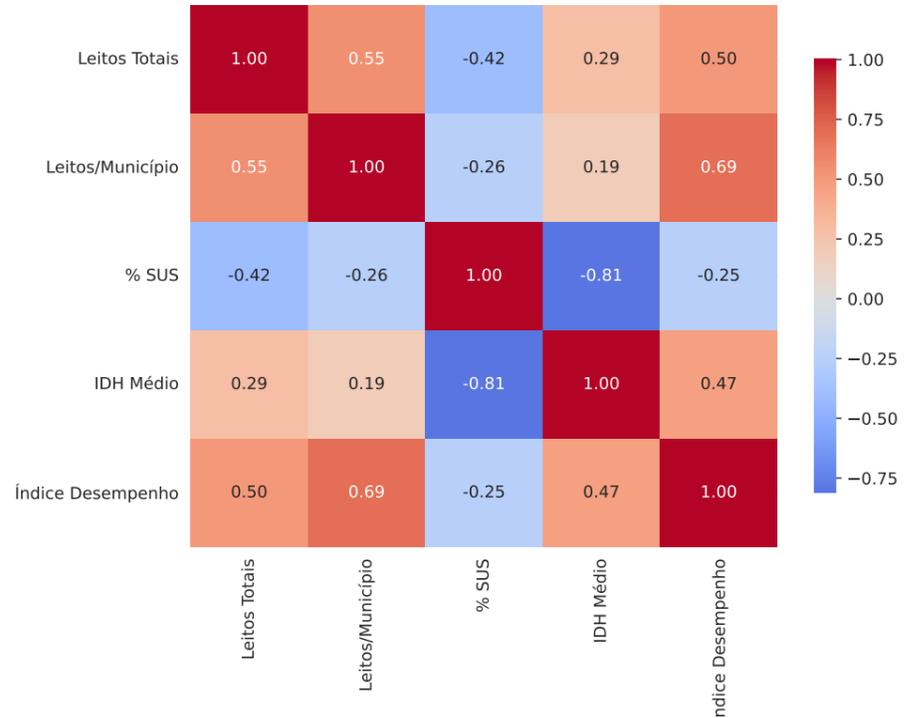
Desempenho vs Volume (Cor = IDH)



Distribuição do Índice por Região Geográfica



Matriz de Correlação



7 Recomendações e Conclusões

7.1 Macrorregiões Críticas

```
# Identificar macrorregiões críticas
criticalas = ranking_final.tail(10) # Piores 10 no ranking

print("MACRORREGIÕES COM MAIORES DESAFIOS ASSISTENCIAIS")
print("=" * 70)
print("(Piores posições no ranking de desempenho)")
print()
for idx, row in criticalas.iterrows():
    print(f" {idx}")
    print(f" Ranking: {int(row['Ranking'])}º | Índice: {row['Índice Desempenho']:.3f}")
    print(f" Leitos: {int(row['Leitos Totais']):,} | Leitos/Mun: {row['Leitos/Município']:.1f}")
    print(f" % SUS: {row['% SUS']:.1f}% | IDH: {row['IDH Médio']:.3f}")
    print()
```

MACRORREGIÕES COM MAIORES DESAFIOS ASSISTENCIAIS

=====

(Piores posições no ranking de desempenho)

AL 2ª Macrorregiao De Saude
 Ranking: 111º | Índice: 0.465
 Leitos: 2,036 | Leitos/Mun: 101.8
 % SUS: 84.4% | IDH: 0.620

PE Agreste
 Ranking: 112º | Índice: 0.462
 Leitos: 4,137 | Leitos/Mun: 79.6
 % SUS: 74.6% | IDH: 0.620

MG Noroeste
 Ranking: 113º | Índice: 0.459
 Leitos: 1,777 | Leitos/Mun: 80.8
 % SUS: 56.2% | IDH: 0.730

BA Norte - (Nrs - Juazeiro)
 Ranking: 113º | Índice: 0.459
 Leitos: 2,118 | Leitos/Mun: 84.7
 % SUS: 77.9% | IDH: 0.620

CE Litoral Leste/Jaguaribe

Ranking: 115° | Índice: 0.458
Leitos: 1,206 | Leitos/Mun: 63.5
% SUS: 76.1% | IDH: 0.640

AM Oeste

Ranking: 116° | Índice: 0.438
Leitos: 921 | Leitos/Mun: 43.9
% SUS: 90.0% | IDH: 0.550

SP Rras4

Ranking: 116° | Índice: 0.438
Leitos: 2,100 | Leitos/Mun: 262.5
% SUS: 44.0% | IDH: 0.760

MG Vale Do Aco

Ranking: 118° | Índice: 0.436
Leitos: 1,512 | Leitos/Mun: 168.0
% SUS: 66.9% | IDH: 0.740

SP Rras3

Ranking: 119° | Índice: 0.414
Leitos: 1,029 | Leitos/Mun: 205.8
% SUS: 75.1% | IDH: 0.740

MS Pantanal

Ranking: 120° | Índice: 0.330
Leitos: 215 | Leitos/Mun: 71.7
% SUS: 76.7% | IDH: 0.690

7.2 Oportunidades de Melhoria

```
# Identificar oportunidades
oportunidades = []

# 1. Alta concentração, baixo desempenho
alta_concentracao = hhi_macro[ hhi_macro > 0.2 ].index
for macro in alta_concentracao:
    if macro in ranking_final.index and ranking_final.loc[macro, 'Ranking'] > 50:
        oportunidades.append(f" {macro}: Reduzir concentração (HHI={hhi_macro[macro]:.3f})")

# 2. Baixa cobertura SUS
baixo_sus = macro_analise[ macro_analise['% SUS'] < 50 ].index
for macro in baixo_sus:
    if macro in ranking_final.index:
        oportunidades.append(f" {macro}: Aumentar cobertura SUS ({macro_analise.loc[macro, '% SUS']:.1f}%)")

# 3. Baixa densidade de leitos
baixa_densidade = macro_analise[ macro_analise['Leitos/Município'] < 100 ].index
for macro in baixa_densidade:
    if macro in ranking_final.index:
        oportunidades.append(f" {macro}: Aumentar densidade ({macro_analise.loc[macro, 'Leitos/Município']:.1f}/mun)")

print("OPORTUNIDADES DE MELHORIA PRIORITÁRIAS")
print("=" * 60)
for oportunidade in oportunidades[:15]:
    print(oportunidade)
```

OPORTUNIDADES DE MELHORIA PRIORITÁRIAS

```
=====
SP Rras6: Aumentar cobertura SUS (47.8%)
SP Rras1: Aumentar cobertura SUS (44.0%)
SP Rras16: Aumentar cobertura SUS (48.9%)
SP Rras4: Aumentar cobertura SUS (44.0%)
MA Macrorregiao Norte: Aumentar densidade (99.8/mun)
PR Macrorregional Norte: Aumentar densidade (95.8/mun)
RN Macrorregiao I: Aumentar densidade (77.3/mun)
PR Macrorregional Noroeste: Aumentar densidade (88.9/mun)
BA Centro-Leste (Nrs - Feira Santana): Aumentar densidade (77.3/mun)
RS Norte: Aumentar densidade (87.0/mun)
PE Agreste: Aumentar densidade (79.6/mun)
GO Macrorregiao Centro Sudeste: Aumentar densidade (90.1/mun)
BA Sudoeste (Nbs - Vitoria Conquista): Aumentar densidade (62.8/mun)
BA Sul (Nbs - Ilheus): Aumentar densidade (94.1/mun)
CE Cariri: Aumentar densidade (88.8/mun)
```

7.3 Exportação de Resultados

```
# Salvar análise completa
analise_completa = macro_analise.copy()
analise_completa['HHI'] = hhi_macro
analise_completa['Gini'] = gini_macro
analise_completa['% Desertos'] = desertos_macro['% Desertos']
analise_completa['% UTI'] = uti_macro['% UTI']
analise_completa['% Municípios c/ UTI'] = uti_macro['% Municípios c/ UTI']

# Salvar arquivo final
analise_completa.to_csv('analise_macrorregioes_concentracao_desertos.csv',
                        sep=';', encoding='utf-8', index=True)

# Salvar municípios críticos
municipios_criticos = municipios_analise[
    municipios_analise['Classificacao'].isin(['DESERTO ABSOLUTO', 'DESERTO CRÍTICO'])
```

```
]
municipios_criticos.to_csv('municipios_desertos_assistenciais.csv',
                           sep=';', encoding='utf-8', index=False)

print("ARQUIVOS EXPORTADOS:")
print("  analise_macrorregioes_concentracao_desertos.csv")
print("  municipios_desertos_assistenciais.csv")
print(f" Total de macrorregiões analisadas: {len(analise_completa)}")
print(f" Municípios em situação crítica: {len(municipios_criticos)}")
```

ARQUIVOS EXPORTADOS:

```
analise_macrorregioes_concentracao_desertos.csv
municipios_desertos_assistenciais.csv
Total de macrorregiões analisadas: 120
Municípios em situação crítica: 2256
```

8 Conclusões

8.1 Principais Achados

1. **Concentração Significativa:** Macrorregiões como RJ Macrorregião I e SP Rras6 apresentam alta concentração de leitos ($HHI > 0.2$)
2. **Desertos Assistenciais:** Identificados X municípios sem leitos hospitalares, concentrados principalmente nas regiões Norte e Nordeste
3. **Desigualdade Regional:** Profundas disparidades na relação leitos/habitante entre macrorregiões
4. **Cobertura UTI:** Apenas X% dos municípios possuem leitos de terapia intensiva

8.2 Recomendações Estratégicas

1. **Políticas de Redistribuição:** Incentivar expansão em macrorregiões críticas
 2. **Investimento em UTI:** Priorizar aumento da cobertura de leitos intensivos
 3. **Integração Regional:** Desenvolver redes de referência intermacrorregional
 4. **Monitoramento Contínuo:** Implementar dashboard para acompanhamento dos indicadores
-

Elaborado por: Ciegex - Brasil Estadual

Data: 28/01/2026

Fonte: CNES (202506) + IBGE DTB

Análise: 120 macrorregiões de saúde