

ANÁLISE DOS IMPACTOS SOCIOECONÔMICOS CAUSADOS PELO COVID – 19 NOS PAÍSES ESCANDINAVOS

Gabriel Araújo de Azevedo Lima

RESUMO

Este artigo faz parte da avaliação de conclusão do curso **Fundamentos Matemáticos e Estatísticos para Ciência de Dados** da UFPB, e tem por finalidade a análise dos **impactos socioeconômicos do COVID – 19** e o questionamento a respeito da **eficiência e necessidade do isolamento social** com base na análise dos países escandinavos, utilizando métodos que usufruíram de dados de diversas naturezas para o estudo. O objetivo primordial é **verificar** as **consequências** e **mudanças** que as medidas protetivas de **isolamento** (ou **não-isolamento**) causaram na estrutura social e econômica dos países nórdicos.

INTRODUÇÃO

A partir do momento em que iniciou-se a circulação do novo coronavírus na Europa, o seu grande potencial de transmissão fez com que, em um pequeno espaço de tempo, o contingente de pessoas infectadas se multiplicasse em grande escala. Devido ao fato de esse ser um vírus mortal e altamente infeccioso, a necessidade de impor políticas governamentais para a suspensão de atividades sociais e econômicas foi reconhecida.

Todavia, era claro aos líderes políticos e à toda população que o *lockdown* seria um grande problema para a economia nacional, de modo a fazer com que todos os setores comerciais e econômicos tivessem que ser limitados. Em consequência, serviços não poderiam mais ser prestados e os países seriam obrigados a passar por um momento de retração econômica.

Nesse contexto, um dos 5 países nórdicos tomou a decisão de não promover medidas rentes as diretrizes comuns do isolamento social. A Suécia propôs que o fechamento total da economia e serviços não faria parte do seu plano, pois seu objetivo era tentar alcançar a chamada “imunidade de rebanho”, a qual sugere que: Ao longo do tempo, o contingente populacional chegaria a um percentual de pessoas imunizadas que faria com que a circulação do vírus fosse contida.

O objeto de estudo para este artigo é a relação entre a estratégia tomada pelo governo sueco para combater o Covid-19 , seus impactos e compará-lo com o desenvolver das ações do vírus nos países escandinavos vizinhos.

PROBLEMA

Visando tal situação, levando em conta a grande discórdia que uma decisão mal aplicada poderia causar em tempos como esse, é necessário confirmar se essa estratégia foi satisfatória e trouxe resultados expressivos em favor da resistência e combate ao Covid – 19. De forma análoga, os indicadores utilizados como ferramentas para auxiliar nesse processo foram: IDH, PIB per capita, taxa de crescimento do PIB, densidade populacional, dados de pirâmides etárias, dados de mortes diárias por coronavírus em cada país, mortes por idade na Dinamarca, mortes por idade na Noruega e mortes por idade na Suécia

METODOLOGIA

O método utilizado para a análise do objeto de estudo é a manipulação dos dados recolhidos e a visualização gráfica deles por meio das bibliotecas da linguagem Python.

DIFICULDADES E SOLUÇÕES

A maior dificuldade encontrada para produzir este artigo foi a trabalhosa caminhada que é a pesquisa e separação de dados, encontrar dados condizentes com o período atual e /ou que sejam compatíveis entre si. A solução para isso foi atentar-se mais à seleção dos dados utilizados e, encontrar boas fontes para estruturar a pesquisa. Infelizmente, nem

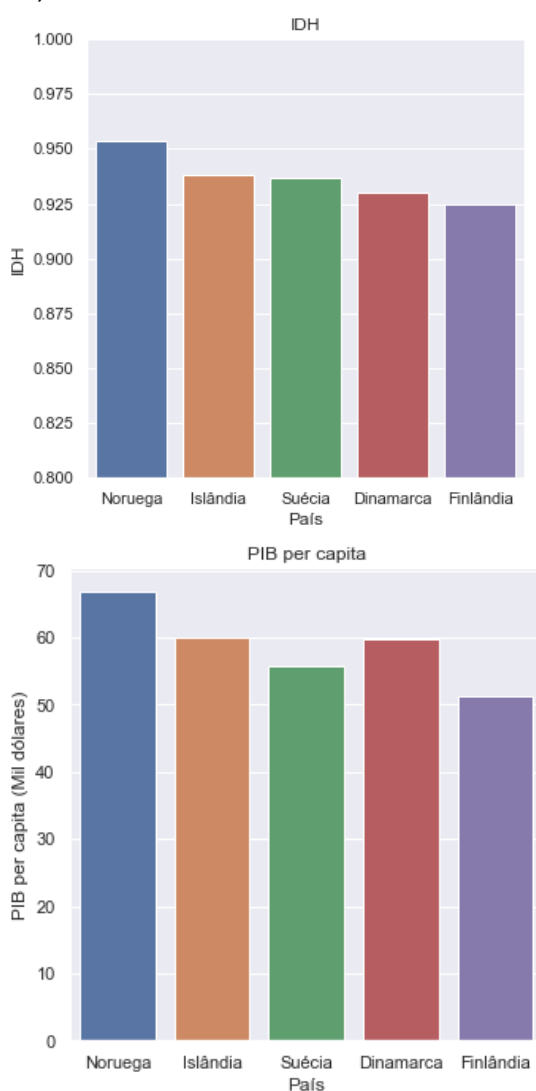
todos os dados que se deseja são encontrados, dessa forma, alguns tiveram que ser colocados de fora da análise.

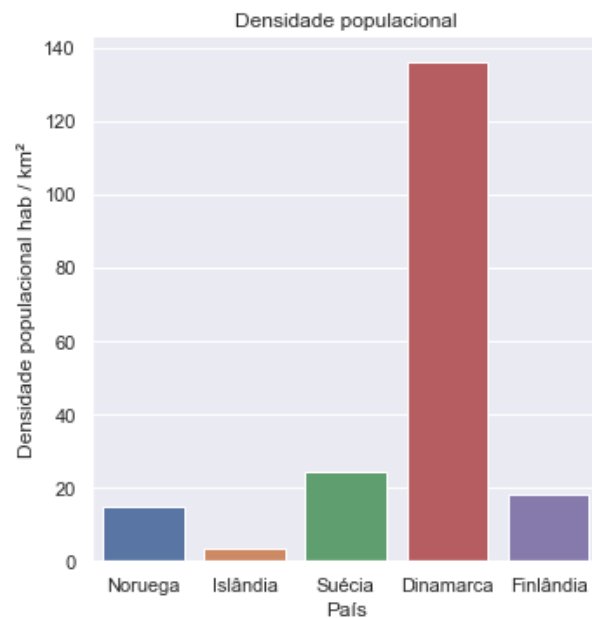
DESENVOLVIMENTO

Para verificar a eficiência dessa estratégia, é necessário comparar os dados recolhidos da Suécia com outros países que escolheram fazer um estrito isolamento social. Comparar diferentes países é uma tarefa árdua, pois cada um é regido por políticas independentes e soberanas, e além disso, têm traços sociais paralelos.

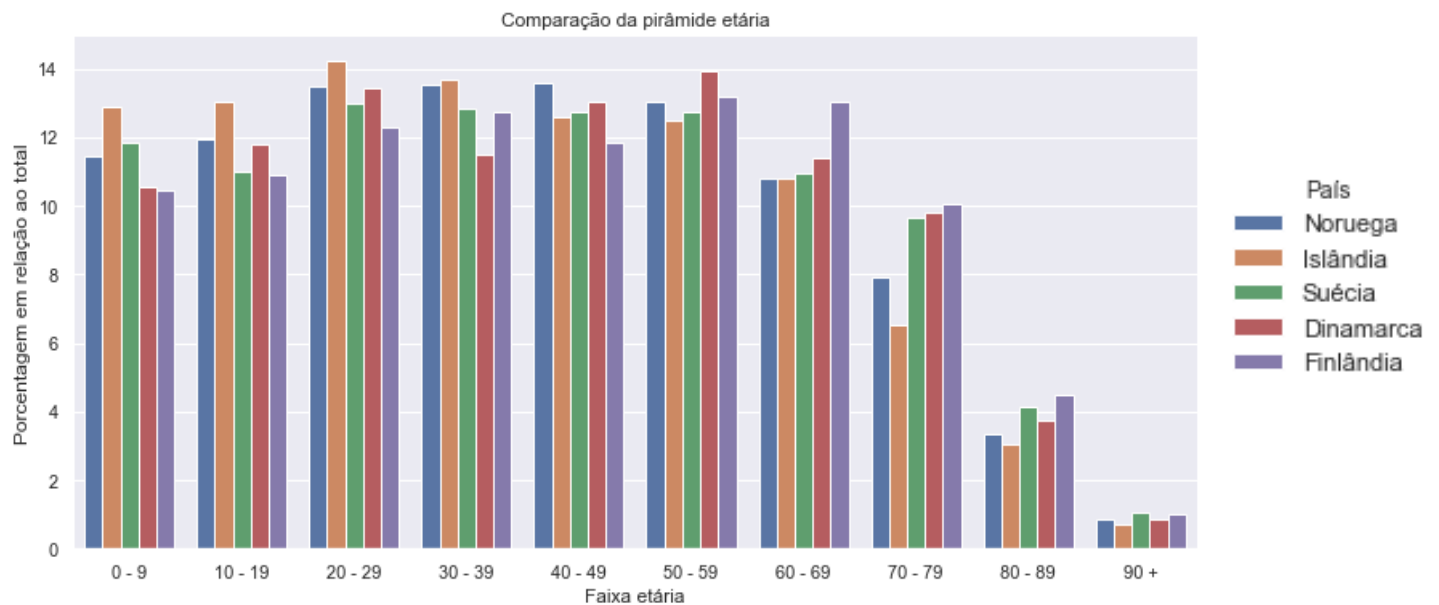
Portanto, para produzir um comparativo o mais fiel possível, há de se comparar países similares em estrutura política e social. Nesse sentido, o conjunto dos países nórdicos possui muitas semelhanças, e isso é verificável tanto na proximidade ou pouca disparidade entre indicadores socioeconômicos quanto na distribuição da pirâmide etária de cada país.

Abaixo estão algumas estatísticas socioeconômicas de cada país (dados de 2019):





País	IDH	PIB per capita (Dólares)	Densidade populacional hab / Km²
Noruega	0.954	66.832	14.726
Islândia	0.938	60.061	3.382
Suécia	0.937	55.815	24.459
Dinamarca	0.930	59.830	136.033
Finlândia	0.925	51.324	18.204



	Noruega	%N	Islândia	%I	Suécia	%S	Dinamarca	%D	Finlândia	%F
0 - 9	616.24K	11.46%	43.66K	12.88%	1.19M	11.87%	607.86K	10.53%	578.8K	10.46%
10 - 19	643.04K	11.96%	44.26K	13.06%	1.10M	11.02%	679.99K	11.78%	602.75K	10.90%
20 - 29	724.42K	13.47%	48.23K	14.23%	1.30M	13.00%	774.99K	13.43%	678.64K	12.27%
30 - 39	727.72K	13.53%	46.46K	13.70%	1.28M	12.85%	662.57K	11.48%	705.21K	12.75%
40 - 49	730.8K	13.59%	42.62K	12.57%	1.27M	12.73%	752.09K	13.03%	655.32K	11.85%
50 - 59	701.45K	13.04%	42.27K	12.47%	1.28M	12.76%	803.94K	13.93%	728.97K	13.18%
60 - 69	581.79K	10.82%	36.63K	10.81%	1.09M	10.93%	657.18K	11.39%	720.69K	13.03%
70 - 79	427.14K	7.94%	22.22K	6.55%	967.44K	9.64%	566.94K	9.82%	556.20K	10.05%
80 - 89	179.11K	3.33%	10.25K	3.03%	416.22K	4.15%	216.58K	3.75%	248.84K	4.50%
90 +	47.10K	0.88%	2.38K	0.70%	105.87K	1.05%	49.69K	0.86%	56.69K	1.02%
Total	5.37M	100.00%	339.03K	100.00%	10.03M	100.00%	5.77M	100.00%	5.53M	100.00%

Tabela com a distribuição da população em cada faixa etária

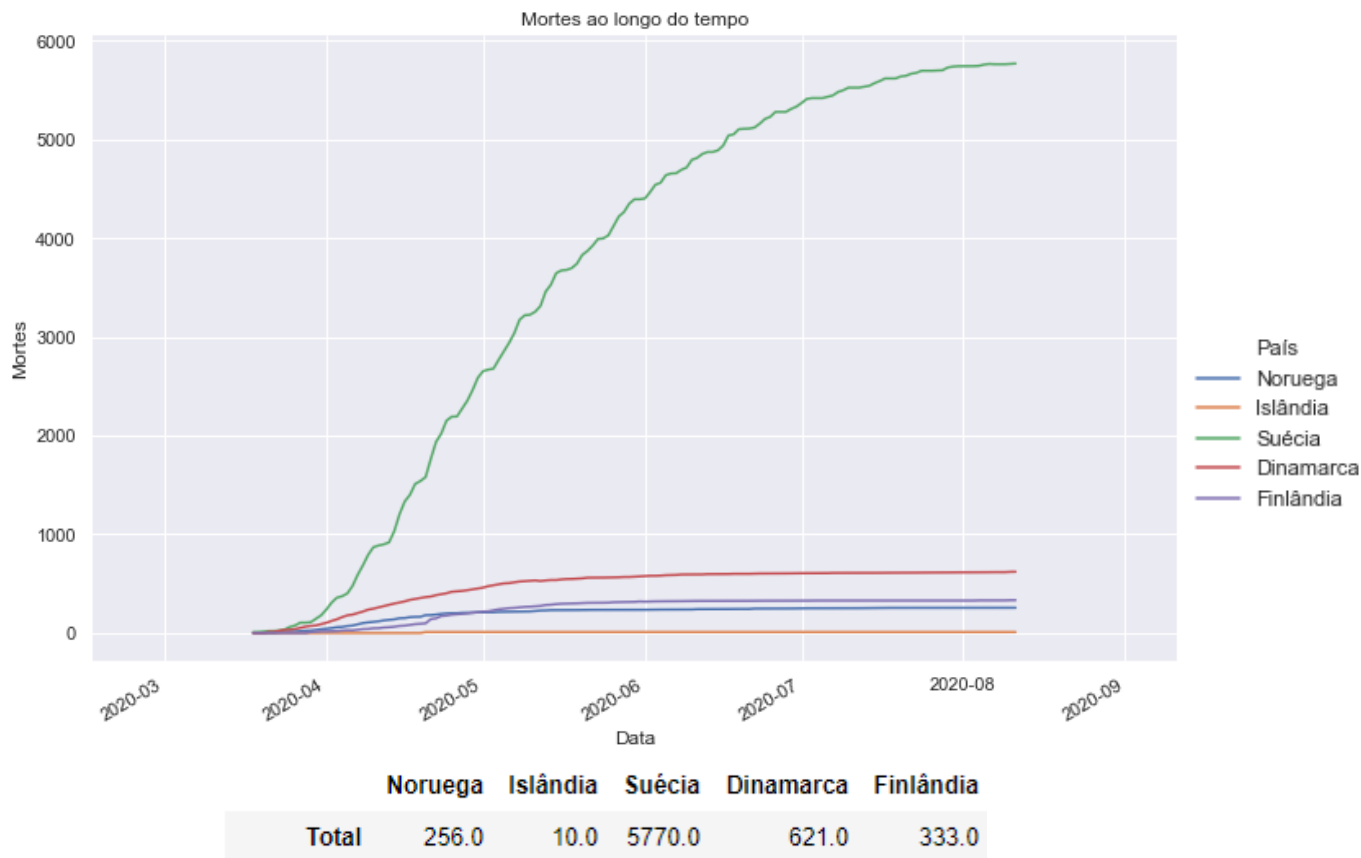
Diante da visualização gráfica e da leitura tabelar de tais indicadores, nota-se uma grande semelhança da estrutura socioeconômica dos países, fato que permite uma comparação mais sucinta dos impactos do covid-19 entre esses ambientes.

Os países nórdicos são notáveis pelo elevado índice de desenvolvimento humano (IDH) apontando para uma ótima qualidade de vida dos nativos da região. O PIB é reflexo da capacidade produtiva de cada país e da relevância de seus serviços em um contexto global.

Ao analisar a distribuição populacional no comparativo das pirâmides etárias, infere-se que esses países possuem pirâmides extremamente parecidas e com fortes traços de consolidação, as quais mostram que a variação entre o percentual de pessoas acima de 60 anos é de aproximadamente 21 a 28% por cento (enquanto a do Brasil é de aproximadamente 13,4%).

Visto que a comparação entre esses países é possível, portas se abriram para a análise da ação do coronavírus em função das práticas adotadas por cada nação.

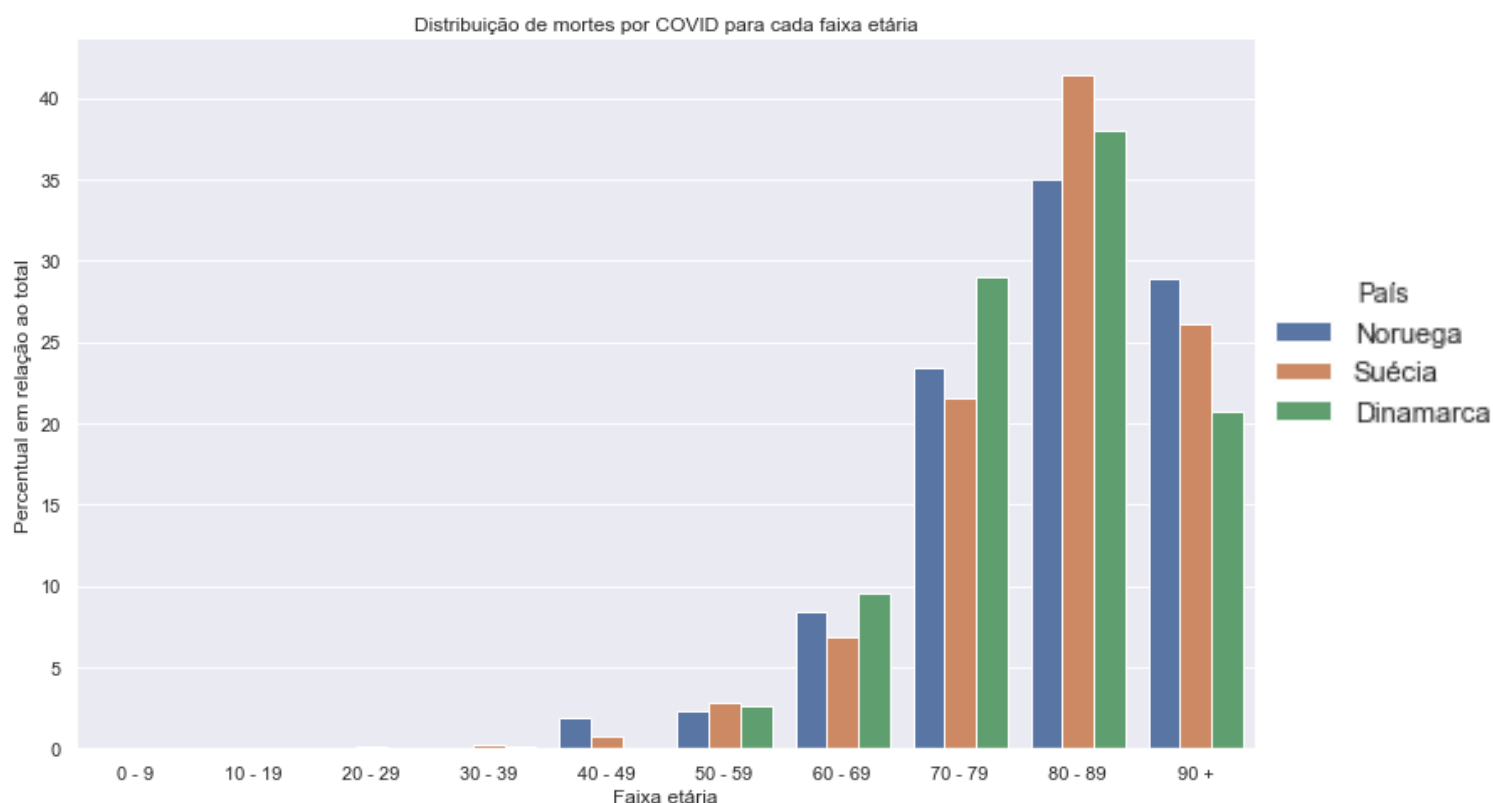
O gráfico abaixo mostra a quantidade de mortes causadas pelo Covid – 19 ao longo de aproximados 4 meses:



Tal gráfico inicia-se no dia 18 de março de 2020 e termina no dia 11 de agosto de 2020. Sob uma análise clara, nota-se que a Suécia possui a curva mais expressiva dos 5 países, apesar da população sueca ser constituída por 10 milhões de pessoas, o número de mortos foi desproporcional em relação aos outros países, os quais, com exceção da Islândia, possuem em média, 5,55 milhões de habitantes, a Suécia teve 0,57% da sua população morta pelo coronavírus. A Islândia foi capaz de conter o número de mortes com excelência, mantendo as mortes em 10 do início ao fim do período analisado, acometeu-se de apenas 0,03% de mortos em relação à população total. Os outros países obtiveram resultados satisfatórios, mantendo controlado o seu contingente de pessoas falecidas. (Noruega: 0,047%. Dinamarca: 0,1%. Finlândia: 0,06%)

Outro objeto de grande importância para a análise é a distribuição de mortes pelo covid-19 entre faixas etárias, pois a partir da visualização gráfica desses dados, é possível visualizar qual faixa teve o maior número de mortes registrados:

O gráfico abaixo mostra a distribuição de mortes pelo coronavírus (percentualmente) nas diferentes faixas etárias de cada país (dados insuficientes para inserir a Islândia e Dinamarca na comparação):



	Noruega	%N	Suécia	% S	Dinamarca	%D
0 - 9	0	0.00	1	0.02	0	0.00
10 - 19	0	0.00	0	0.00	0	0.00
20 - 29	0	0.00	9	0.16	0	0.00
30 - 39	0	0.00	16	0.28	1	0.16
40 - 49	5	1.95	44	0.76	0	0.00
50 - 59	6	2.34	161	2.79	16	2.58
60 - 69	22	8.59	398	6.89	59	9.50
70 - 79	60	23.44	1245	21.56	180	28.99
80 - 89	88	34.38	2396	41.50	236	38.00
90 +	75	29.30	1504	26.05	129	20.77
Total	256	100.00	5774	100.00	621	100.00

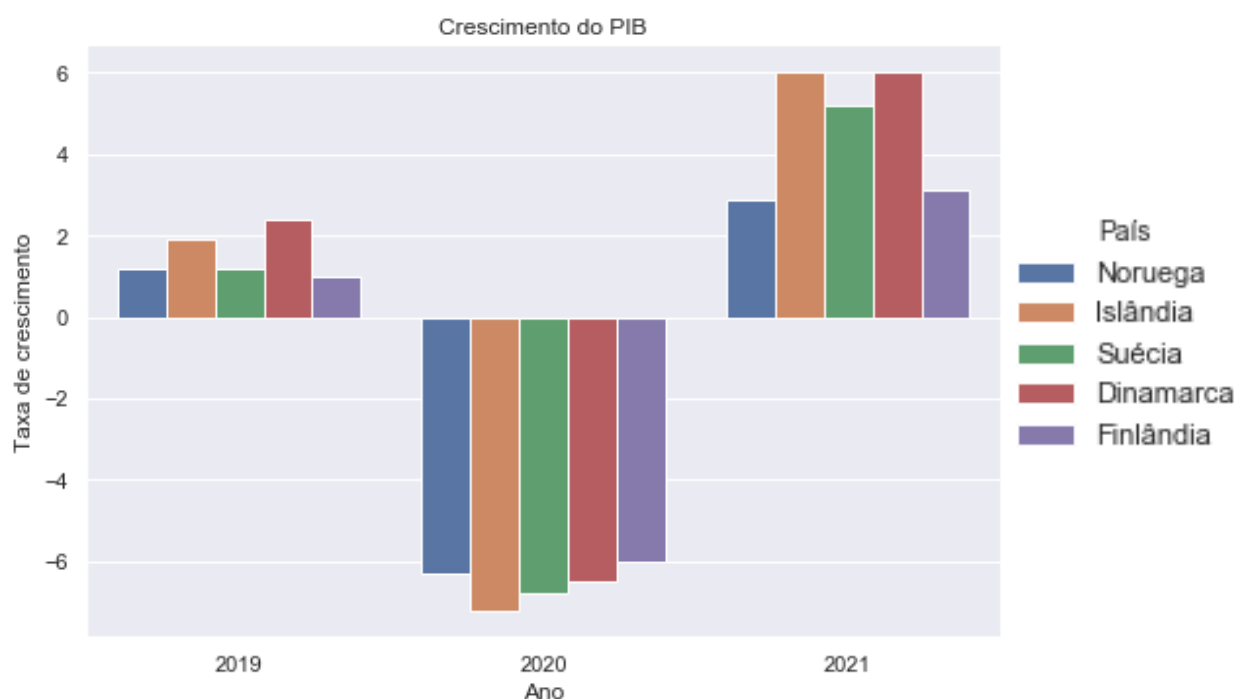
DataFrame com os dados quantitativos e percentuais

Em termos gerais, as pessoas idosas são as mais propensas a serem vítimas fatais do coronavírus, a variação do percentual de mortos

com mais de 60 anos vai de 95,7 a 97,3% do contingente total de vítimas, ou seja, as taxas são próximas.

Apesar da proximidade desse valor percentual, mediante a análise da quantidade de mortos com mais de 60 anos em relação à população total dessa mesma faixa etária, resultados discrepantes são identificados. Nesse contexto, a Noruega perdeu 0,2% de sua população acima dos 60 anos, e a Dinamarca perdeu o dobro, 0,4%. No caso da Suécia, a população com 60 anos ou mais foi reduzida em 2%, dez vezes mais do que a Noruega e cinco vezes mais do que a Dinamarca, um número desproporcional se comparado aos registrados pelas nações vizinhas.

Além de aspectos relacionados à saúde pública, vale ressaltar a análise da taxa de crescimento do PIB no ano de 2020 e a projeção para o ano de 2021, com o objetivo de inferir os resultados das estratégias tomadas pelos países na esfera econômica.



A queda do PIB em 2020 foi iminente a todos os países nórdicos, com a limitação do comércio e indústria aliada ao pessimismo do mercado, projeções de queda entre 6 e 7,2% foram tangíveis. No entanto, as projeções para 2021 são otimistas, sendo muito maiores percentualmente do que aquelas do ano de 2019, fato necessário para que a economia seja reestruturada.

CONCLUSÕES

De acordo com o analisado neste artigo, é possível concluir que, as atitudes do governo sueco se tornaram ineficientes para o combate ao coronavírus, de modo a caracterizar o isolamento como uma necessidade indubitável e imprescindível. A almejada “imunidade de rebanho” proposta como solução pela Suécia nunca foi confirmada, e as escassas práticas de isolamento resultaram em um número de mortos desproporcional se comparado ao número dos vizinhos nórdicos.

A curva de mortes no país sueco continuou a subir até quando a dos outros países nórdicos encaminhou-se à uma estabilização. Mesmo com uma população maior em relação a dos outros países, o valor absoluto das mortes, se racionado ao da população total, é extremamente superior. Tal fato poderia ter sido evitado caso o país tenha seguido diretrizes semelhantes àquelas fundamentadas na Noruega, por exemplo.

Em relação à especulação sobre o descuido do governo sueco em proteger os idosos e dar suporte à asilos e afins, baseado na concentração de mortos com mais de 60 anos, uma conclusão sucinta não pode ser tirada, o percentual de mortos nessa faixa etária na Suécia é de 96%, enquanto na Noruega e Dinamarca são, 95,7% e 97,3%, respectivamente. Entretanto, em consonância a dados fornecidos pela matéria *Coronavirus: What's going wrong in Sweden's care homes?* Da *BBC News*, 48,9% das mortes pelo covid-19 em idosos aconteceram em asilos e casas de cuidado. Além disso, o número de vítimas acima de 60 anos na Suécia foi, percentualmente, muito maior do que o número analisado nos outros países comparados, logo, é possível concluir que o descuido do governo sueco com os ancestrais é realidade.

Tratando-se da esfera econômica, os esforços do governo sueco não foram o suficiente para frear o brusco fechamento do PIB projetado para 2020. A taxa de queda foi semelhante a dos outros países, porém, foi

a segunda maior queda percentual, ficando atrás somente da Islândia, fato que não deveria ter sido esperado pela Suécia.

A estratégia ajudou a otimizar a projeção para 2021, entretanto, a taxa não chegou a superar o crescimento da Islândia e Dinamarca. O crescimento da Suécia está projetado para ser de 5,2%, enquanto a média de crescimento entre os outros países é de 4,5%. É notável que existe uma pequena vantagem em relação a média, mas continuam próximas. Portanto, o método utilizado para contornar o impasse socioeconômico causado pelo novo coronavírus não foi bem como o governo do país esperaria.

REFERÊNCIAS

<https://tinyurl.com/qk3pkpn> - IDH

<https://tinyurl.com/y967okgt> - PIB per capita

<https://tinyurl.com/yykubkka> - % de crescimento anual do PIB

<https://tinyurl.com/yx9ejsax> - Densidade populacional

<https://www.populationpyramid.net/> - Dados de pirâmides etárias

<https://tinyurl.com/y4ksyohz> - Dados de mortes diárias por coronavírus em cada país

<https://tinyurl.com/y6xc9f5y> - Mortes por idade na Dinamarca

<https://tinyurl.com/yyqrulng> - Mortes por idade na Noruega

<https://tinyurl.com/y9u5g89j> - Mortes por idade na Suécia

<https://www.bbc.com/news/world-europe-52704836> - Matéria da BBC

APÊNDICE

Códigos implementados:

In[1]:

```
#Importando bibliotecas.  
import numpy as np  
import pandas as pd  
import seaborn as sns  
import matplotlib  
import matplotlib.pyplot as plt
```

In[2]:

```
#Criando arrays com os nomes dos países para auxiliar na limpeza de dados e  
montagem de dataframes.
```

```
países = np.array(['Noruega', 'Islândia', 'Suécia', 'Dinamarca', 'Finlândia'])  
países_ingles = np.array(['Norway', 'Iceland', 'Sweden', 'Denmark', 'Finland'])
```

```
#Criando as Series dos indicadores socioeconômicos.
```

```
idh = pd.Series([0.954, 0.938, 0.937, 0.930, 0.925], index = países, name = 'IDH')
```

```
ppc = pd.Series([66.832, 60.061, 55.815, 59.830, 51.324], index = países, name = 'PIB  
per capita (Mil dólares)')
```

```
dp = pd.read_csv('densidade.csv')
```

In[3]:

```
#Limpando densidade.csv para encontrar densidade populacional. Dados de 2019.
```

```
dp = dp[['Location', 'PopDensity']].loc[dp['MidPeriod'] == 2019.5].rename(columns =  
{ 'Location': 'País' })
```

```
dp = dp.set_index('País')
```

```
dp = dp.loc[['Norway', 'Iceland', 'Sweden', 'Denmark', 'Finland']]
```

In[4]:

```
#Traduzindo os nomes do csv da densidade populacional
```

```
dp = dp.reset_index()
```

```
dp['País'] = países
```

```
dp = dp.set_index('País').rename(columns = {'PopDensity': 'Densidade populacional hab / km²'})
```

```
# In[5]:
```

```
#Criando o dataframe para leitura de tabela de todos os indicadores.
```

```
indicadores = pd.DataFrame({  
'IDH': idh,  
'PIB per capita (Dólares)':ppc,  
'Densidade populacional hab / Km²':dp['Densidade populacional hab / km²']})
```

```
indicadores_leitura = indicadores.copy()
```

```
# In[6]:
```

```
#Aplicando sufixos para melhorar a leitura
```

```
for i in range(indicadores_leitura.shape[0]):
```

```
    indicadores_leitura.iloc[i,1] = f'{indicadores_leitura.iloc[i,1]:.2f}K'
```

```
    indicadores_leitura.iloc[i,2] = f'{indicadores_leitura.iloc[i,2]:.1f}'
```

```
# In[7]:
```

```
#Criando o dataframe para a pirâmide etária
```

```
pop_noruega = pd.read_csv ('pop_noruega.csv')  
pop_dinamarca = pd.read_csv('pop_dinamarca.csv')  
pop_suecia = pd.read_csv ('pop_suecia.csv')  
pop_finlandia = pd.read_csv('pop_finlandia.csv')  
pop_islandia = pd.read_csv ('pop_islandia.csv')
```

```
# In[8]:
```

```
#Limpando os dados e consertando índices da pirâmide etária.
```

```
#Criando uma função para automatizar a soma dos sexos.
```

```
def t_piramide(d):
```

```
    d['Total'] = d['M'] + d['F']
```

```
    df = d[['Age', 'Total']].set_index('Age')
```

```
    return df
```

```
piramide = pd.DataFrame({  
'Noruega': t_piramide( pop_noruega)['Total'],  
'Islândia': t_piramide( pop_islandia)['Total'],  
'Suécia': t_piramide( pop_suecia)['Total'],
```

```
'Dinamarca': t_piramide(pop_dinamarca)['Total'],
'Finlândia': t_piramide(pop_finlandia)['Total'])
piramide.iloc[-2] += piramide.iloc[-1]
piramide = piramide.drop(['100+'])
```

```
# In[9]:
```

```
#Ajustando classes
```

```
for i in range((piramide.shape[0] - 10)):
    piramide.iloc[i] += piramide.iloc[i+1]
    piramide = piramide.drop([f'{list(piramide.index)[i + 1]}'])
```

```
piramide.loc['Total'] = piramide.sum()
```

```
# In[10]:
```

```
#Criando novos índices para condizer com as novas classes.
```

```
novo_indice_piramide = []
```

```
n1 = 0
```

```
n2 = 9
```

```
for i in range(piramide.shape[0] - 1):
    if i == list(range(piramide.shape[0] - 1))[-1] :
        novo_indice_piramide.append(f'{n1} +')
```

```
    else:
        novo_indice_piramide.append(f'{n1} - {n2}')
```

```
    n1 += 10
```

```
    n2 += 10
```

```
novo_indice_piramide.append('Total')
```

```
piramide.index = novo_indice_piramide
```

```
# In[11]:
```

```
#Construindo percentuais das idades para ajudar na leitura e discussão.
```

```
def serie_percentual_leitura(c):
```

```
    lista_de_percentuais = []
```

```
    for i in range(piramide.shape[0]):
```

```
        lista_de_percentuais.append(f'{(piramide.iloc[i,c]/piramide.iloc[-1,c]) * 100 :.2f}%')
```

```
    s = pd.Series(lista_de_percentuais, index = novo_indice_piramide)
```

```

    return s

def serie_percentual(c):
    lista_de_percentuais = []

    for i in range(piramide.shape[0]):
        lista_de_percentuais.append(float(f'{(piramide.iloc[i,c]/piramide.iloc[-1,c]) * 100 :.2f}'))

    s = pd.Series(lista_de_percentuais, index = novo_indice_piramide)

    return s

```

In[12]:

#Criando um dataframe percentual para integrar na tabela de leitura.

```

percentual_leitura = pd.DataFrame({
'%N': serie_percentual_leitura(0),
'%I': serie_percentual_leitura(1),
'%S': serie_percentual_leitura(2),
'%D': serie_percentual_leitura(3),
'%F': serie_percentual_leitura(4)})

```

In[13]:

#Criando um dataframe da pirâmide etária para leitura.

```

piramide_sufixo = piramide.copy()

```

In[14]:

#Aplicando sufixos K e M, parte 1.

```

for j in range(piramide_sufixo.shape[1]):
    for i in range(piramide_sufixo.shape[0]):
        if len(str(piramide_sufixo.iloc[i,j])) >= 4 and len(str(piramide_sufixo.iloc[i,j])) <= 6:
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j] / 1000)
        elif len(str(piramide_sufixo.iloc[i,j])) >= 7:
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j] / 1000000)

```

In[15]:

#Aplicando sufixos K e M, parte 2.

```

for j in range(piramide_sufixo.shape[1]):
    for i in range(piramide_sufixo.shape[0]):

        if float(piramide_sufixo.iloc[i, j]) >= 100 and len( str(piramide_sufixo.iloc[i,j])) >= 7:
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j])[:-1] + 'K'

        elif float(piramide_sufixo.iloc[i, j]) >= 100 and len(str(piramide_sufixo.iloc[i,j])) == 6
        :
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j]) + 'K'

        elif float(piramide_sufixo.iloc[i, j]) >= 100 and len(str(piramide_sufixo.iloc[i,j])) == 5
        :
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j]) + 'K'

        elif float(piramide_sufixo.iloc[i, j]) >= 10 and len(str(piramide_sufixo.iloc[i,j])) == 6
        :
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j])[:-1] + 'K'

        elif float(piramide_sufixo.iloc[i, j]) >= 10 and len(str(piramide_sufixo.iloc[i,j])) == 5
        :
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j]) + 'K'

        elif float(piramide_sufixo.iloc[i, j]) >= 10 and len(str(piramide_sufixo.iloc[i,j])) == 6
        :
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j])[:-1] + 'K'

        elif float(piramide_sufixo.iloc[i, j]) >= 1 and len(str(piramide_sufixo.iloc[i,j])) == 5 :
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j])[:-1] + 'K'

        elif float(piramide_sufixo.iloc[i, j]) >= 1 and len(str(piramide_sufixo.iloc[i,j])) == 7 :
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j])[:-3] + 'M'

        elif len(str(piramide_sufixo.iloc[i,j])) >= 8:
            piramide_sufixo.iloc[i,j] = str(piramide_sufixo.iloc[i,j])[:-4] + 'M'

```

In[16]:

#Integrando percentual ao dataframes.

```

indice_percentual = list(range(1, 11, 2))

```

```

dict_percentual = zip(indice_percentual, list(percentual_leitura))

```

```

for i, (k, v) in enumerate(dict_percentual):

```

```

    piramide_sufixo.insert(k, v, percentual_leitura[v])

```

In[17]:

#Criando um dataframe percentual para criar o gráfico de comparação das pirâmides.

```

piramide_percentual = pd.DataFrame({
    'Noruega': serie_percentual(0),
    'Islândia': serie_percentual(1),
    'Suécia': serie_percentual(2),

```



```
'Dinamarca': serie_percentual(3),
'Finlândia': serie_percentual(4))
piramide_percentual.drop(['Total'])
```

```
# In[18]:
```

```
#Iniciando trabalho com as mortes por data
```

```
mortes_tempo = pd.read_csv('mortes_tempo.csv')
```

```
mortes_tempo = mortes_tempo[['Date', 'Country', 'Deaths']]
```

```
mortes_tempo = mortes_tempo.loc[mortes_tempo['Deaths'] >= 10].set_index('Date')
```

```
# In[19]:
```

```
#Criando uma função para limpar os dados
```

```
def clean_mortes_pais (n,nf):
```

```
    df = mortes_tempo.loc[mortes_tempo['Country'] == f'{n}'].rename(columns = {'Deaths':
f'{nf}'})
```

```
    df = df[f'{nf}']
```

```
    return df
```

```
# In[20]:
```

```
#Criando um DataFrame das mortes por país ao longo do tempo
```

```
mortes_pais_tempo = pd.DataFrame({
'Noruega' : clean_mortes_pais ( 'Norway', 'Noruega'),
'Islândia' : clean_mortes_pais ( 'Iceland', 'Islândia'),
'Suécia' : clean_mortes_pais ( 'Sweden', 'Suécia'),
'Dinamarca': clean_mortes_pais ('Denmark', 'Dinamarca'),
'Finlândia': clean_mortes_pais ('Finland', 'Finlândia')})
```

```
mortes_pais_tempo = mortes_pais_tempo.fillna(0)
```

```
# In[21]:
```

```
#Iniciando trabalho com mortes por faixa etária.
```

```
mortes_noruega = np.array([ 0, 0, 0, 0, 5, 6, 22, 60, 88, 75])
```

```
mortes_suecia = np.array([1, 0, 9, 16, 44, 161, 398, 1245, 2396, 1504])
```

```
mortes_dinamarca = np.array([ 0, 0, 0, 1, 0, 16, 59, 180, 236, 129])
```

```
mortes_por_covid = pd.DataFrame({
'Noruega' : mortes_noruega,
'Suécia' : mortes_suecia,
'Dinamarca': mortes_dinamarca},
```

```
index = novo_indice_piramide[:-1:1])
mortes_por_covid.loc['Total'] = mortes_por_covid.sum()
```

In[22]:

#Criando percentuais para criar grafico comparativo da distribuição de mortes por faixa etária em cada país.

```
mortes_percentuais = []
```

```
def morte_percentual_numero(c):
```

```
    mortes_percentuais = []
```

```
    for i in range(11):
```

```
        mortes_percentuais.append((float(f'{(mortes_por_covid.iloc[i,c]/mortes_por_covid.iloc[-1,c]) * 100 :.2f}')))
```

```
    s = pd.Series(mortes_percentuais, index = novo_indice_piramide)
```

```
    return s
```

In[23]:

```
morte_percentual = pd.DataFrame({
'%N' : morte_percentual_numero(0),
'%S' : morte_percentual_numero(1),
'%D' : morte_percentual_numero(2)})
```

In[24]:

#Integrando percentual à tabela.

```
mortes_por_covid_leitura = mortes_por_covid.copy()
```

```
indice_morte_percentual = list(range(1, 7, 2))
```

```
dict_morte_percentual = zip(indice_morte_percentual, list(morte_percentual))
```

```
for i, (k, v) in enumerate(dict_morte_percentual):
```

```
    mortes_por_covid_leitura.insert(k, v, morte_percentual[v])
```

In[25]:

#Importando taxa de crescimento do PIB.

```
taxa_pib = pd.read_excel('taxa_pib.xlsx').fillna(0)
```

```
taxa_pib = taxa_pib.rename(columns = {'Real GDP growth (Annual percent change)':
'Taxa de crescimento anual do PIB'}).set_index('Taxa de crescimento anual do PIB')
```

```
# In[26]:
```

```
taxa_pib = taxa_pib['Denmark':'Sweden']  
taxa_pib = taxa_pib[[2019, 2020, 2021]]
```

```
# In[27]:
```

```
taxa_pib = taxa_pib.T[['Norway', 'Iceland', 'Sweden', 'Denmark', 'Finland']]
```

```
# In[28]:
```

```
taxa_pib = taxa_pib.rename(columns = dict(zip(paises_ingles, paises)))
```

```
# In[29]:
```

```
#Resetando indexes para criação de gráficos com o seaborn
```

```
piramide_percentual_reset =  
piramide_percentual.drop(['Total']).reset_index().rename(columns = {'index': 'Faixa  
etária'})  
mortes_por_covid_reset =  
mortes_por_covid.drop(['Total']).reset_index().rename(columns = {'index': 'Faixa etária'})  
mortes_pais_tempo_reset = mortes_pais_tempo.reset_index().rename(columns =  
{'index': 'Data'})  
taxa_pib_reset = taxa_pib.reset_index().rename(columns = {'index': 'Ano'})  
morte_percentual_reset =  
morte_percentual.drop(['Total']).reset_index().rename(columns = {'index': 'Faixa  
etária', '%N': 'Noruega', '%S': 'Suécia', '%D': 'Dinamarca'})
```

```
# In[30]:
```

```
#fazendo o melting dos dataframes necessários
```

```
piramide_melt = pd.melt(piramide_percentual_reset, 'Faixa etária', var_name =  
'País', value_name = 'Porcentagem em relação ao total')  
mortes_por_covid_melt = pd.melt(mortes_por_covid_reset, 'Faixa etária', var_name =  
'País', value_name = 'Mortes')  
mortes_pais_tempo_melt = pd.melt(mortes_pais_tempo_reset, 'Data', var_name =  
'País', value_name = 'Mortes')  
taxa_pib_melt = pd.melt(taxa_pib_reset, 'Ano', var_name = 'País',  
value_name = 'Taxa de crescimento')  
morte_percentual_melt = pd.melt(morte_percentual_reset, 'Faixa etária', var_name =  
'País', value_name = 'Percentual em relação ao total')
```

```
# In[31]:
```

```
#Aplicando meses.
```

```
meses = {'01':'Janeiro', '02':'Fevereiro', '03':'Março', '04':'Abril', '05':'Maio', '06':'Junho',
'07':'Julho', '08':'Agosto', '09':'Setembro', '10':'Outubro', '11':'Novembro', '12':'Dezembro'}
```

```
# In[32]:
```

```
coluna_mes = []
```

```
# In[33]:
```

```
for i in range(mortes_pais_tempo_melt.shape[0]):
    coluna_mes.append(meses[mortes_pais_tempo_melt.iloc[i, 0].replace('-',
').strip().split()[1]])
```

```
# In[34]:
```

```
#Definindo estilo de gráfico
sns.set(style = 'darkgrid')
```

```
# In[35]:
```

```
mortes_pais_tempo_melt['Data'] = pd.to_datetime(mortes_pais_tempo_melt['Data'])
```

```
# In[36]:
```

```
#Criando gráfico de linhas (Mortes por tempo)
```

```
grafico1 = sns.relplot('Data', 'Mortes', hue = 'País', kind = 'line', data =
mortes_pais_tempo_melt)
grafico1.fig.autofmt_xdate()
grafico1.fig.set_size_inches(15,7)
grafico1.set(title = 'Mortes ao longo do tempo')
grafico1.ax.set_xlim((mortes_pais_tempo_melt.reset_index()['Data'].min() -
pd.DateOffset(days=31)),(mortes_pais_tempo_melt.reset_index()['Data'].max() +
pd.DateOffset(days=31)))
```

```
# In[37]:
```

```
#Criando gráfico de barras (Distribuição da pirâmide)
```

```
grafico2 = sns.catplot('Faixa etária', 'Porcentagem em relação ao total', kind = 'bar', hue
= 'País', data = piramide_melt)
grafico2.fig.set_size_inches(15,5)
grafico2.set(title = 'Comparação da pirâmide etária')
```

```
# In[38]:
```

#Gráfico de barras (IDH)

```
grafico3 = sns.catplot('País', 'IDH', kind = 'bar', data = idh.reset_index().rename(columns = {'index': 'País'}))  
grafico3.fig.set_size_inches(5,5)  
grafico3.set(ylim = (0.8, 1))  
grafico3.set(title = 'IDH')
```

In[39]:

#Gráfico de barras (PIB per capita)

```
grafico4 = sns.catplot('País', 'PIB per capita (Mil dólares)', kind = 'bar', data =  
ppc.reset_index().rename(columns = {'index': 'País'}))  
grafico4.fig.set_size_inches(5,5)  
grafico4.set(title = 'PIB per capita')
```

In[40]:

#Gráfico de barras (Densidade populacional)

```
grafico5 = sns.catplot('País', 'Densidade populacional hab / km²', kind = 'bar', data =  
dp.reset_index().rename(columns = {'index': 'País'}))  
grafico5.fig.set_size_inches(5,5)  
grafico5.set(title = 'Densidade populacional')
```

In[41]:

```
morte_percentual_melt = morte_percentual_melt.rename(columns = {'%N': 'Noruega',  
'%S': 'Suécia', '%D': 'Dinamarca'})
```

In[42]:

#Gráfico de barras (Distribuição de mortes ao longo da idade)

```
grafico6 = sns.catplot('Faixa etária', 'Percentual em relação ao total', kind = 'bar', hue =  
'País', data = morte_percentual_melt)  
grafico6.fig.set_size_inches(15,7)  
grafico6.set(title = 'Distribuição de mortes por COVID para cada faixa etária')
```

In[43]:

#Gráfico de barras (Crescimento do PIB)

```
grafico7 = sns.catplot('Ano', 'Taxa de crescimento', kind = 'bar', hue = 'País', data =  
taxa_pib_melt)  
grafico7.fig.set_size_inches(10, 5)  
grafico7.set(title = 'Crescimento do PIB')
```

