

# PSL Paper

April 28, 2025

## 1 Economic complexity and international trade: A case study on the State of Goiás (2010-2019)

Artigo publicado no PSL Quarterly sobre a economia do estado de Goiás e sua relação com o meio ambiente. Dados usados foram os da COMEX, para o cálculo do Índice de Herfindahl-Hirschman (HHI), que possui a seguinte fórmula:

$$HHI = \sum_{i=1}^n s_i^2$$

Em que  $s_i$  é a participação das exportações do produto de categoria  $i$  no total das exportações do estado.

**Links:** [Link para o artigo](#)

[Dados da COMEX](#)

Obs:

A base usada para a versão do artigo possui manipulações próprias que não foram de minha autoria

## 2 Importando os pacotes e definindo alguns parâmetros

```
[1]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
C:\Users\joaop\anaconda3\lib\site-packages\pandas\core\arrays\masked.py:60:
UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version
'1.3.5' currently installed).
from pandas.core import (
```

```
[2]: plt.style.use('ggplot')
plt.rcParams['font.family'] = 'Cambria'
```

```
[3]: valor = 'Valor FOB (US$)'
ano = 'Ano'
bloco = 'Bloco Econômico Destino'
```

```
[4]: caminho_arquivo = r"Gráficos PSL.xlsx"
```

### 3 Importando os dados

```
[5]: df = pd.read_excel(r"Tabelas - Projeto MESP 2022.09.02.xlsx", sheet_name = 'Dados Brutos - Comex')

blocos = ['Mercado Comum do Sul - Mercosul',
          'Comunidade Andina das Nações - CAN',
          'União Europeia - UE',
          'Associação de Nações do Sudeste Asiático - ASEAN']

df = df[~df[bloco].isin(blocos)]

df
```

```
[5]:
```

	Código SH2	Descrição SH2	Código SH4 \
0	1	Animais vivos	101
1	1	Animais vivos	101
2	1	Animais vivos	101
4	1	Animais vivos	101
6	1	Animais vivos	101
...	...	...	...
73043	99	Transações especiais	9998
73044	99	Transações especiais	9998
73045	99	Transações especiais	9998
73046	99	Transações especiais	9998
73047	99	Transações especiais	9998

		Descrição SH4	Código SH6 \
0		Cavalos, asininos e muares, vivos	10111
1		Cavalos, asininos e muares, vivos	10121
2		Cavalos, asininos e muares, vivos	10190
4		Cavalos, asininos e muares, vivos	10190
6		Cavalos, asininos e muares, vivos	10190
...		...	...
73043	Consumo de bordo (combustíveis e lubrificantes...		999802
73044	Consumo de bordo (combustíveis e lubrificantes...		999802
73045	Consumo de bordo (combustíveis e lubrificantes...		999802
73046	Consumo de bordo (combustíveis e lubrificantes...		999802
73047	Consumo de bordo (combustíveis e lubrificantes...		999802

		Descrição SH6	Código NCM \
0		Cavalos reprodutores, de raça pura	1011100
1		Cavalos reprodutores de raça pura	1012100
2	Animais vivos das espécies cavalar, asinina e ...		1019010

4	Animais vivos das espécies cavalar, asinina e ...	1019010
6	Animais vivos das espécies cavalar, asinina e ...	1019010
...	...	...
73043	Consumo de bordo para embarcações e aeronaves ...	99980201
73044	Consumo de bordo para embarcações e aeronaves ...	99980201
73045	Consumo de bordo para embarcações e aeronaves ...	99980201
73046	Consumo de bordo para embarcações e aeronaves ...	99980201
73047	Consumo de bordo para embarcações e aeronaves ...	99980201

	Descrição NCM \
0	Cavalos reprodutores, de raça pura
1	Cavalos reprodutores de raça pura
2	Cavalos vivos, exceto reprodutores raça pura
4	Cavalos vivos, exceto reprodutores raça pura
6	Cavalos vivos, exceto reprodutores raça pura
...	...
73043	Consumo de bordo - qualquer outra mercadoria p...
73044	Consumo de bordo - qualquer outra mercadoria p...
73045	Consumo de bordo - qualquer outra mercadoria p...
73046	Consumo de bordo - qualquer outra mercadoria p...
73047	Consumo de bordo - qualquer outra mercadoria p...

	Código - Classificação Tecnológica \
0	PP
1	PP
2	PP
4	PP
6	PP
...	...
73043	Other
73044	Other
73045	Other
73046	Other
73047	Other

	Descrição - Classificação Tecnológica	País Destino \
0	Produtos Primários	Estados Unidos
1	Produtos Primários	Estados Unidos
2	Produtos Primários	Uruguai
4	Produtos Primários	Argentina
6	Produtos Primários	Argentina
...	...	...
73043	Outros	Venezuela
73044	Outros	Venezuela
73045	Outros	Líbano
73046	Outros	Venezuela
73047	Outros	Líbano

	Bloco Econômico Destino	Valor FOB (US\$)	Quilograma Líquido \
0	América do Norte	1800	420
1	América do Norte	12500	450
2	América do Sul	46500	1000
4	América do Sul	7000	500
6	América do Sul	9000	1500
...	...	...	...
73043	América do Sul	77913	341820
73044	América do Sul	231887	829280
73045	Oriente Médio	279606	1671980
73046	América do Sul	215317	764620
73047	Oriente Médio	500	200000

	Quantidade Estatística	Ano \
0	1	2000
1	1	2013
2	2	2010
4	1	2009
6	3	2008
...	...	...
73043	341820	2012
73044	829280	2009
73045	1671980	2008
73046	764620	2008
73047	200000	2004

	Fonte
0	<a href="http://comexstat.mdic.gov.br/pt/geral/130949">http://comexstat.mdic.gov.br/pt/geral/130949</a>
1	<a href="http://comexstat.mdic.gov.br/pt/geral/98650">http://comexstat.mdic.gov.br/pt/geral/98650</a>
2	<a href="http://comexstat.mdic.gov.br/pt/geral/107870">http://comexstat.mdic.gov.br/pt/geral/107870</a>
4	<a href="http://comexstat.mdic.gov.br/pt/geral/111146">http://comexstat.mdic.gov.br/pt/geral/111146</a>
6	<a href="http://comexstat.mdic.gov.br/pt/geral/113880">http://comexstat.mdic.gov.br/pt/geral/113880</a>
...	...
73043	<a href="http://comexstat.mdic.gov.br/pt/geral/101576">http://comexstat.mdic.gov.br/pt/geral/101576</a>
73044	<a href="http://comexstat.mdic.gov.br/pt/geral/110291">http://comexstat.mdic.gov.br/pt/geral/110291</a>
73045	<a href="http://comexstat.mdic.gov.br/pt/geral/113016">http://comexstat.mdic.gov.br/pt/geral/113016</a>
73046	<a href="http://comexstat.mdic.gov.br/pt/geral/113054">http://comexstat.mdic.gov.br/pt/geral/113054</a>
73047	<a href="http://comexstat.mdic.gov.br/pt/geral/125106">http://comexstat.mdic.gov.br/pt/geral/125106</a>

[51145 rows x 17 columns]

```
[6]: # Volume total exportado
```

```
df1 = df[['valor', 'ano']].groupby(ano, as_index = False).sum()

fig, ax = plt.subplots(dpi = 720)
```

```

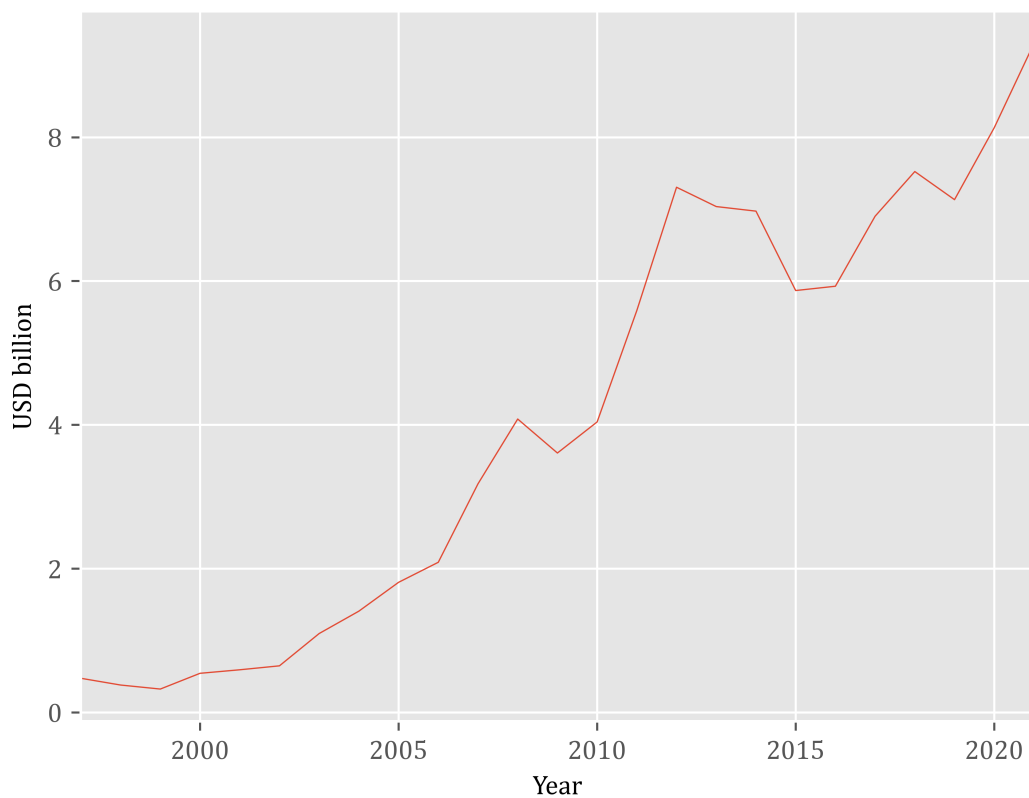
with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
    ↳if_sheet_exists='replace') as writer:
    df1.to_excel(writer, sheet_name='grafico 2', index=False)

ax.plot(df1[ano], df1[valor]/10000000000, linewidth = 0.5)

ax.set(xlabel = 'Year',
      ylabel = 'USD billion',
      xlim = (1997,2021))

```

```
[6]: [Text(0.5, 0, 'Year'), Text(0, 0.5, 'USD billion'), (1997.0, 2021.0)]
```



## 4 HHI Países

```

[7]: # Definindo função do HHI

def HHI(vetor):

    total = np.sum(vetor)

```

```

shares = vetor/total

hhi = np.sum([i**2 for i in shares])

return hhi

```

[8]: # HHI por regiões

```
df1 = df[[ano,bloco,valor]].groupby([ano,bloco], as_index = False).sum()
```

[9]: df2 = df[[ano,valor]].groupby([ano], as\_index = False).sum()

```

df2['HHI'] = 0

for i in df2[ano]:

    df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i,valor])

with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
                    if_sheet_exists='replace') as writer:
    df2.to_excel(writer, sheet_name='grafico 3', index=False)

fig, ax = plt.subplots(dpi = 720)

ax.plot(df2[ano], df2['HHI'], linewidth = 0.8),

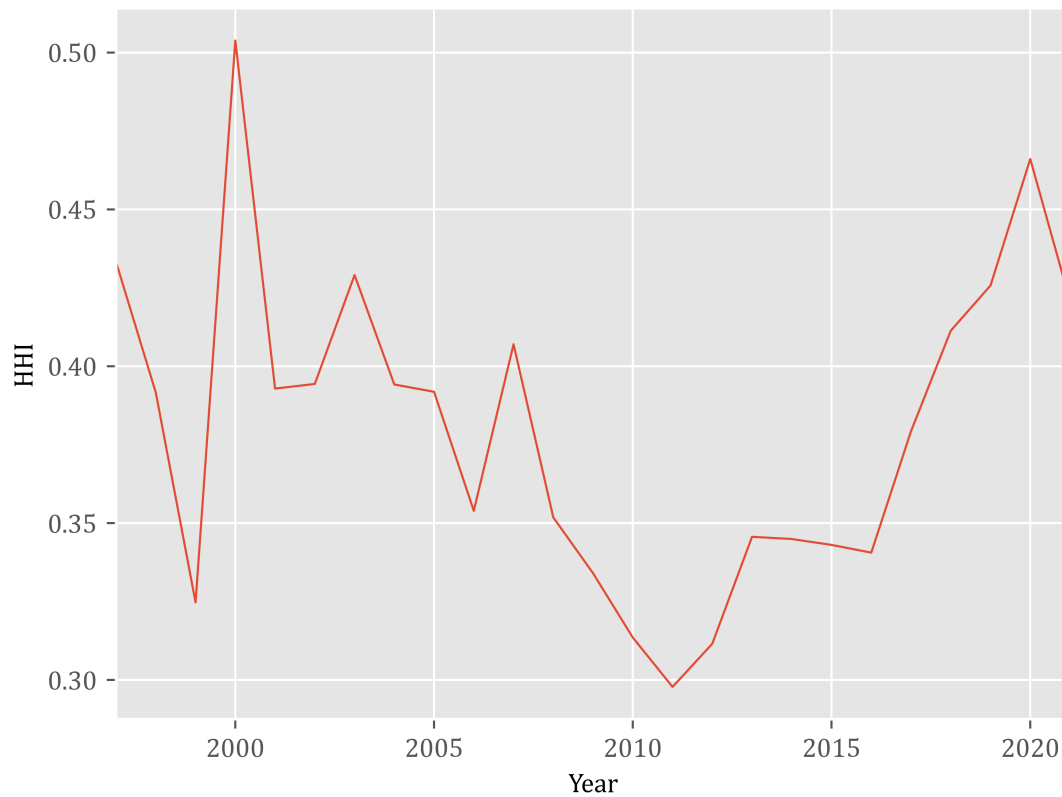
ax.set(xlabel = 'Year',
       ylabel = 'HHI',
       xlim = (1997,2021))

```

C:\Users\joaop\AppData\Local\Temp\ipykernel\_26652\2177638797.py:7:  
FutureWarning: Setting an item of incompatible dtype is deprecated and will  
raise an error in a future version of pandas. Value '0.4332168111819197' has  
dtype incompatible with int64, please explicitly cast to a compatible dtype  
first.

```
df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i,valor])
```

[9]: [Text(0.5, 0, 'Year'), Text(0, 0.5, 'HHI'), (1997.0, 2021.0)]



## 5 Gráfico de Barra

```
[10]: df1 = df[[valor,bloco]].groupby([bloco], as_index = False).sum()

regions_dict = {
    'Central America': 'América Central e Caribe',
    'North America': 'América do Norte',
    'South America': 'América do Sul',
    'Europe': 'Europa',
    'Oceania': 'Oceania',
    'Middle East': 'Oriente Médio',
    'Africa': 'África',
    'Asia': 'Ásia (Exclusive Oriente Médio)'
}

for i in regions_dict:
    df1.loc[df1[bloco] == regions_dict[i],bloco] = i

df1
```

```
[10]: Bloco Econômico Destino Valor FOB (US$)
0      Central America      556011459
1      North America      3920465971
2      South America      3515707932
3      Europe      32900595259
4      Oceania      148415247
5      Middle East      7715614518
6      Africa      4624035816
7      Asia      48616323413
```

```
[11]: df1 = df1.sort_values(by = valor, ascending = False)

with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
                    if_sheet_exists='replace') as writer:
    df1.to_excel(writer, sheet_name='grafico 4', index=False)

fig, ax = plt.subplots(dpi = 720)

ax.bar(df1[bloco], df1[valor]/1000000000, color = 'blue', alpha = 0.5)

ax.set_xticklabels(df1[bloco], rotation = -35, ha = 'left')

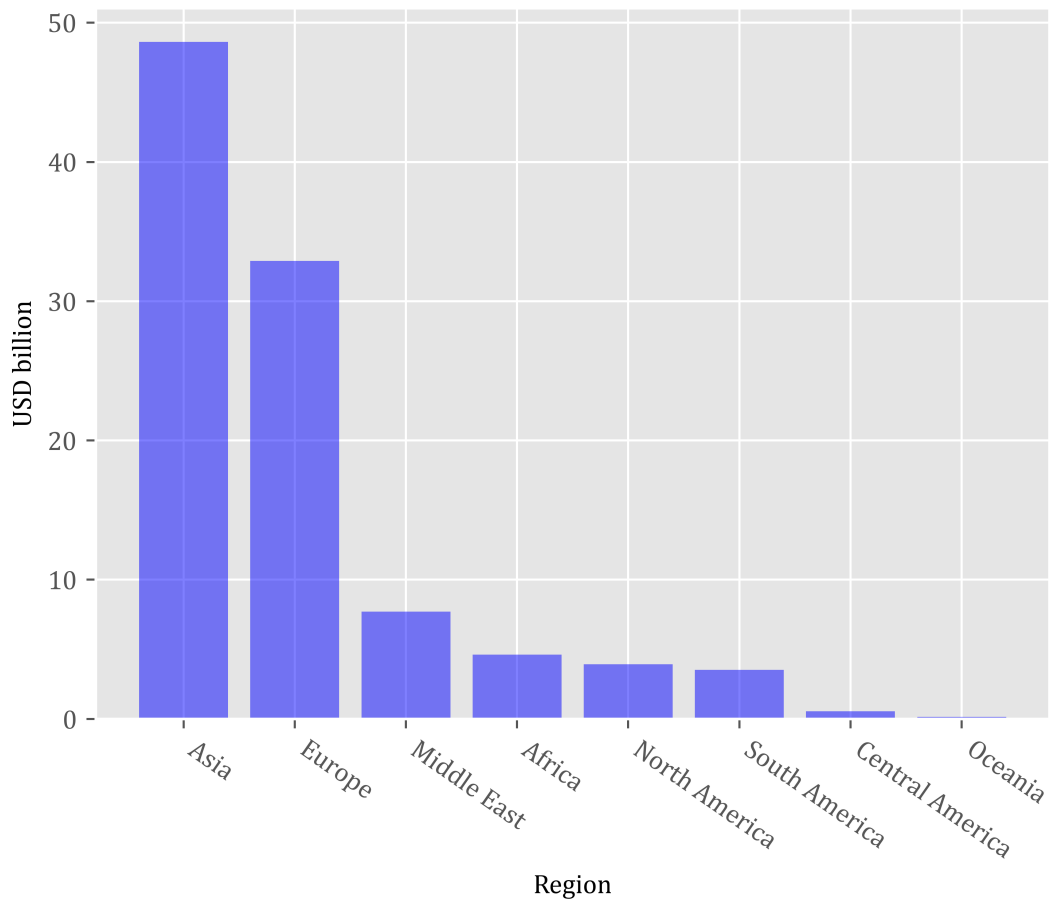
ax.set(xlabel = 'Region',
       ylabel = 'USD billion')
```

C:\Users\joaop\AppData\Local\Temp\ipykernel\_26652\4275275191.py:10: UserWarning:  
FixedFormatter should only be used together with FixedLocator

```
ax.set_xticklabels(df1[bloco], rotation = -35, ha = 'left')
```

```
[11]: [Text(0.5, 0, 'Region'), Text(0, 0.5, 'USD billion')]
```





## 6 Gráfico de Percentuais

```
[12]: df1 = df[[bloco,valor,ano]].groupby([ano,bloco], as_index = False).sum()

blocos = df1[bloco].unique()

blocks = ['Central America',
          'North America',
          'South America',
          'Europe',
          'Oceania',
          'Middle East',
          'Africa',
          'Asia']

dic = dict(zip(blocos,blocks))
```

```

for i in dic:

    df1.loc[df1[bloco] == i, bloco] = dic[i]

df1

```

```

[12]:      Ano Bloco Econômico Destino  Valor FOB (US$)
0    1997      Central America      1032902
1    1997      North America      55902142
2    1997      South America      33896490
3    1997      Europe      298052895
4    1997      Oceania      352953
..    ...
195  2021      Europe      2024250093
196  2021      Oceania      10921621
197  2021      Middle East      603009782
198  2021      Africa      353972276
199  2021      Asia      5624660854

```

[200 rows x 3 columns]

```

[13]: df2 = df1.pivot(index = ano, columns = bloco, values = valor)

totais = df2.sum(axis = 1).values

for i in df2.columns:

    df2[i] = df2[i].values/totais

df2.reset_index()

```

```

[13]: Bloco Econômico Destino  Ano  Africa  Asia  Central America  Europe \
0    1997  0.027160  0.142819      0.002173  0.626928
1    1998  0.017376  0.142225      0.008495  0.586852
2    1999  0.027337  0.150359      0.015134  0.509709
3    2000  0.008646  0.105590      0.008286  0.689378
4    2001  0.024061  0.133329      0.007378  0.595622
5    2002  0.019499  0.123364      0.033870  0.596979
6    2003  0.024930  0.156939      0.002951  0.624716
7    2004  0.039258  0.141494      0.001431  0.595313
8    2005  0.050437  0.209757      0.001089  0.580381
9    2006  0.077824  0.247994      0.003310  0.527628
10   2007  0.048336  0.258492      0.001263  0.576672
11   2008  0.046286  0.374744      0.006004  0.451285
12   2009  0.057263  0.394437      0.006522  0.405427
13   2010  0.055480  0.376170      0.004881  0.393201

```

14	2011	0.052686	0.403569	0.004354	0.340053
15	2012	0.070745	0.450187	0.010664	0.306107
16	2013	0.052711	0.484476	0.006342	0.315213
17	2014	0.059072	0.492106	0.003115	0.301723
18	2015	0.048426	0.501398	0.005634	0.279417
19	2016	0.042574	0.488063	0.005087	0.300686
20	2017	0.032290	0.546699	0.007918	0.265083
21	2018	0.032626	0.592844	0.004590	0.225082
22	2019	0.032162	0.599791	0.003629	0.244233
23	2020	0.033318	0.651851	0.004922	0.186488
24	2021	0.038036	0.604402	0.004627	0.217517

Bloco Econômico Destino	Middle East	North America	Oceania	South America
0	0.011294	0.117585	0.000742	0.071298
1	0.034203	0.145867	0.000819	0.064163
2	0.032360	0.183493	0.000367	0.081240
3	0.017516	0.120430	0.000503	0.049652
4	0.076788	0.097961	0.000100	0.064762
5	0.036502	0.125540	0.000008	0.064239
6	0.053884	0.093801	0.000116	0.042663
7	0.067519	0.106251	0.000120	0.048613
8	0.062273	0.049129	0.000291	0.046644
9	0.073596	0.042477	0.000156	0.027015
10	0.062635	0.026763	0.000179	0.025659
11	0.061434	0.016913	0.005327	0.038007
12	0.100451	0.012614	0.000941	0.022345
13	0.111434	0.018650	0.000957	0.039227
14	0.104732	0.022712	0.001175	0.070719
15	0.084308	0.038897	0.001306	0.037786
16	0.084416	0.023113	0.001035	0.032694
17	0.078500	0.034430	0.001642	0.029411
18	0.094092	0.035282	0.002819	0.032933
19	0.084696	0.041499	0.002445	0.034951
20	0.083706	0.035905	0.001207	0.027192
21	0.076247	0.037884	0.001399	0.029327
22	0.057212	0.037926	0.001222	0.023825
23	0.044754	0.049768	0.001498	0.027402
24	0.064797	0.039022	0.001174	0.030425

```
[14]: blocos = ['North America',
               'South America',
               'Asia',
               'Europe']

# df2 = df2[blocos].reset_index()
```

```

with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
                    if_sheet_exists='replace') as writer:
    df2.to_excel(writer, sheet_name='grafico 5', index=False)

fig, ax = plt.subplots(dpi = 720)

for i in blocos:

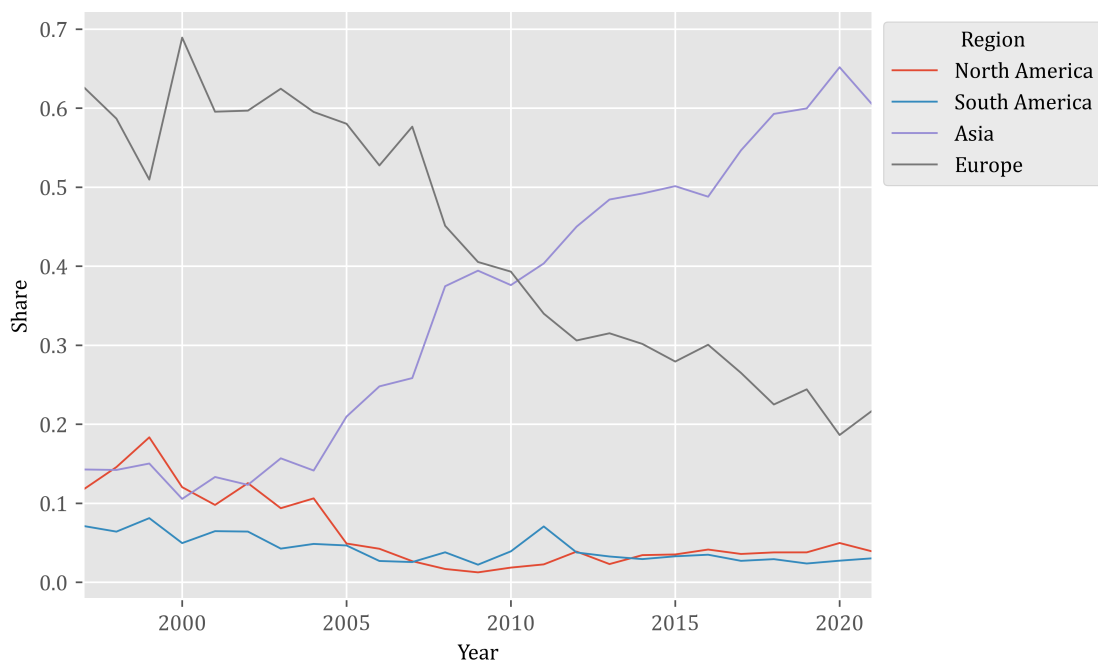
    ax.plot(df2.index, df2[i], label = i, linewidth = 0.85)

ax.legend(bbox_to_anchor = (1,1), loc = 'upper left', title = 'Region')

ax.set(xlim = (1997,2021),
       ylabel = 'Share',
       xlabel = 'Year')

```

```
[14]: [(1997.0, 2021.0), Text(0, 0.5, 'Share'), Text(0.5, 0, 'Year')]
```



## 7 Gráfico de Linha

```

[15]: df1 = df[[bloco,valor,ano]].groupby([ano,bloco], as_index = False).sum()

      blocos = df1[bloco].unique()

```

```

blocks = ['Central America',
          'North America',
          'South America',
          'Europe',
          'Oceania',
          'Middle East',
          'Africa',
          'Asia']

dic = dict(zip(blocos,blocks))

for i in dic:

    df1.loc[df1[bloco] == i, bloco] = dic[i]

df1

```

```

[15]:
   Ano Bloco Econômico Destino  Valor FOB (US$)
0   1997      Central America    1032902
1   1997      North America    55902142
2   1997      South America    33896490
3   1997      Europe    298052895
4   1997      Oceania    352953
..   ...
195  2021      Europe    2024250093
196  2021      Oceania    10921621
197  2021      Middle East    603009782
198  2021      Africa    353972276
199  2021      Asia    5624660854

```

[200 rows x 3 columns]

```

[16]: blocos = ['Asia',
               'Europe']

df2 = df1.pivot(columns = bloco, index = ano, values = valor)[blocos]

for i in blocos:

    coeficientes = np.polyfit(df2.index, df2[i], 10)

    preds = np.polyval(coeficientes, df2.index)

    fit = i + ' ' + 'fit'

```

```

df2[fit] = preds

df2.reset_index()

with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
                    if_sheet_exists='replace') as writer:
    df2.reset_index().to_excel(writer, sheet_name='grafico 6', index=False)

```

C:\Users\joaop\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3460: RankWarning: Polyfit may be poorly conditioned

```
exec(code_obj, self.user_global_ns, self.user_ns)
```

C:\Users\joaop\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3460: RankWarning: Polyfit may be poorly conditioned

```
exec(code_obj, self.user_global_ns, self.user_ns)
```

```

[17]: fig, ax = plt.subplots(dpi = 720)

for i in blocos:

    ax.plot(df2.index, df2[i]/1000000000, label = i, linewidth = 0.85)

    fit = i + ' ' + 'fit'

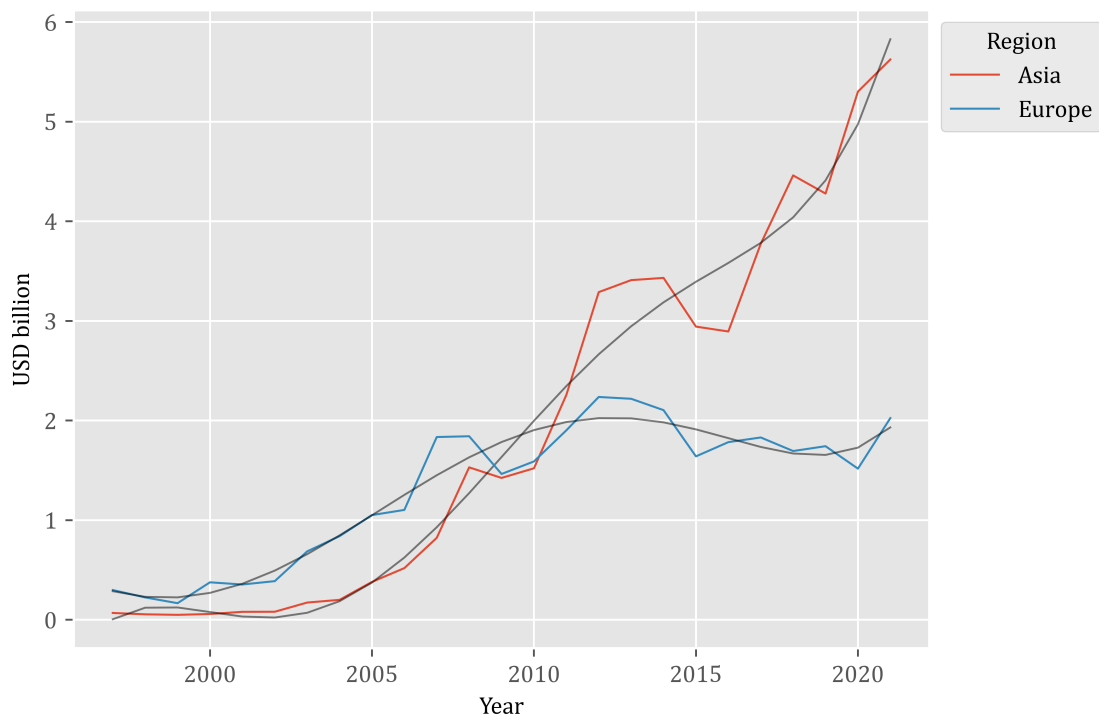
    ax.plot(df2.index, df2[fit]/1000000000, color = 'black', linewidth = 0.8,
            alpha = 0.5)

ax.legend(bbox_to_anchor = (1,1), loc = 'upper left', title = 'Region')

ax.set(xlabel = 'Year',
       ylabel = 'USD billion')

```

```
[17]: [Text(0.5, 0, 'Year'), Text(0, 0.5, 'USD billion')]
```



## 8 HHI Países

```
[18]: pais = 'País Destino'

df1 = df[[pais,valor,ano]].groupby([pais,ano], as_index = False).sum()

df1
```

```
[18]:
```

	País Destino	Ano	Valor FOB (US\$)
0	Afeganistão	2006	263122
1	Afeganistão	2007	150924
2	Afeganistão	2008	247224
3	Afeganistão	2009	111677
4	Afeganistão	2010	131643
...	...	...	...
3309	Índia	2017	390576889
3310	Índia	2018	314789645
3311	Índia	2019	81407297
3312	Índia	2020	106061740
3313	Índia	2021	264418223

[3314 rows x 3 columns]

```
[19]: df2 = df1[['ano', 'valor']].groupby(ano, as_index = False).sum()

df2['HHI'] = 0

for i in df2[ano]:

    df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i, 'valor'])

with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
                    if_sheet_exists='replace') as writer:
    df2.to_excel(writer, sheet_name='grafico 7', index=False)

fig, ax = plt.subplots(dpi = 720)

ax.plot(df2[ano], df2['HHI'], linewidth = 0.8),

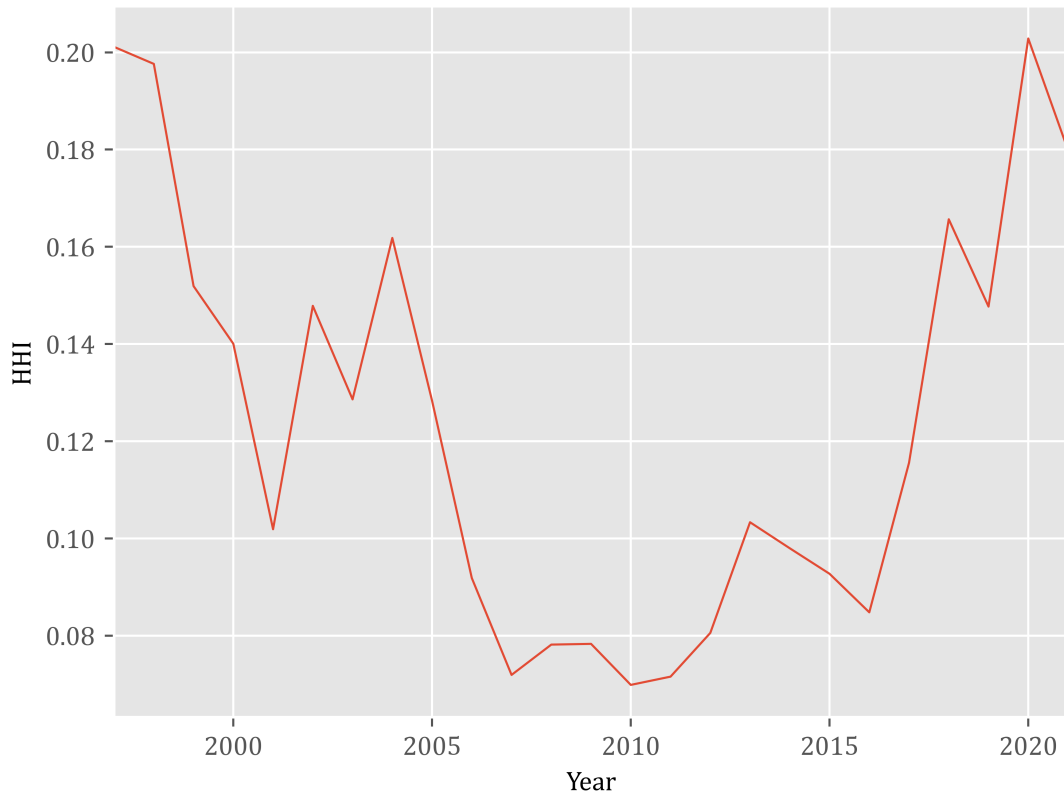
ax.set(xlabel = 'Year',
       ylabel = 'HHI',
       xlim = (1997, 2021))
```

C:\Users\joaop\AppData\Local\Temp\ipykernel\_26652\764432724.py:7: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a future version of pandas. Value '0.201117937940909' has dtype incompatible with int64, please explicitly cast to a compatible dtype first.

```
df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i, 'valor'])
```

```
[19]: [Text(0.5, 0, 'Year'), Text(0, 0.5, 'HHI'), (1997.0, 2021.0)]
```





## 9 Exportações por categoria

```
[20]: produtos = 'Descrição - Classificação Tecnológica'

df1 = df[[produtos, valor, ano]].groupby([ano, produtos], as_index = False).sum()

dicionario = {
    'Manufaturas Baseadas em Recursos Primários': 'Resource-Based Manufactures',
    'Manufaturas de Alta Tecnologia': 'High-Technology Manufactures',
    'Manufaturas de Baixa Tecnologia': 'Low-Technology Manufactures',
    'Manufaturas de Média Tecnologia': 'Medium-Technology Manufactures',
    'Outros': 'Others',
    'Produtos Primários': 'Primary Products'
}

for i in dicionario:
    df1.loc[df1[produtos] == i, produtos] = dicionario[i]
```

```
[21]: df2 = df1.pivot(index = ano, columns = produtos, values = valor)

      totais = df2.sum(axis = 1).values

      for i in df2.columns:

          df2[i] = df2[i].values/totais

      with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
          ↪if_sheet_exists='replace') as writer:
          df2.reset_index().to_excel(writer, sheet_name='grafico 8', index=False)
```

```
[22]: fig, ax = plt.subplots(dpi = 720)

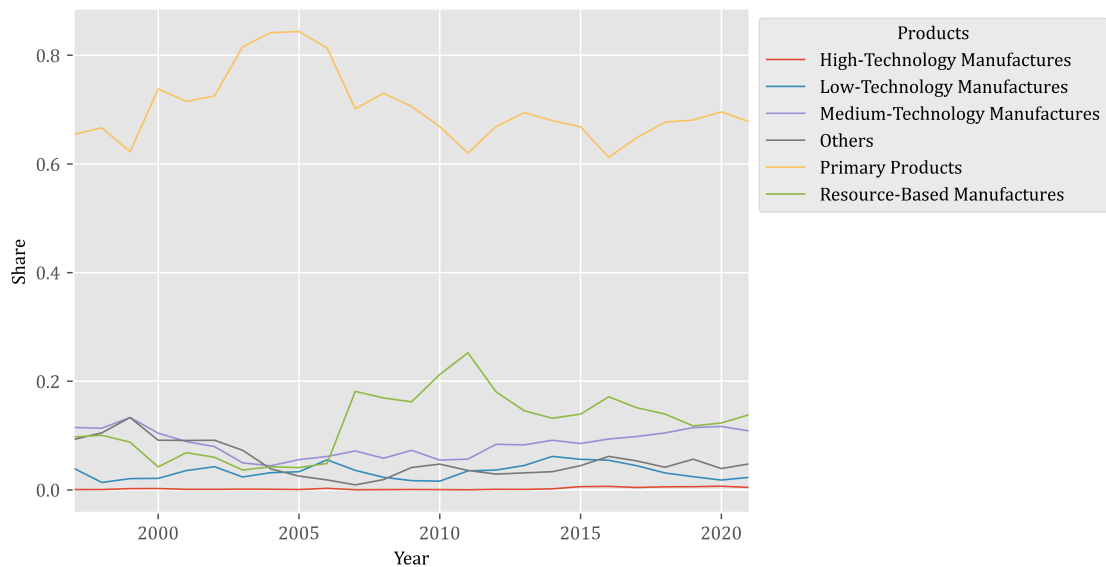
      for i in df2.columns:

          ax.plot(df2.index, df2[i], label = i, linewidth = 0.85)

      ax.legend(bbox_to_anchor = (1,1), loc = 'upper left', title = 'Products')

      ax.set(xlim = (1997,2021),
          ylabel = 'Share',
          xlabel = 'Year')
```

```
[22]: [(1997.0, 2021.0), Text(0, 0.5, 'Share'), Text(0.5, 0, 'Year')]
```



## 10 HHI categorias tecnologia

```
[23]: categoria = 'Código - Classificação Tecnológica'

df1 = df[[categoria,valor,ano]].groupby([categoria,ano], as_index = False).sum()

df1
```

```
[23]:      Código - Classificação Tecnológica  Ano  Valor FOB (US$)
0                                     HT  1997          285166
1                                     HT  1998          242445
2                                     HT  1999          753758
3                                     HT  2000         1349481
4                                     HT  2001          618972
..                                     ...  ...
145                                    RB  2017        1043523937
146                                    RB  2018        1050695545
147                                    RB  2019         840001459
148                                    RB  2020        1001781331
149                                    RB  2021        1290105232
```

[150 rows x 3 columns]

```
[24]: df2 = df1[[ano,valor]].groupby(ano, as_index = False).sum()

df2['HHI'] = 0

for i in df2[ano]:

    df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i,valor])

with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
                    if_sheet_exists='replace') as writer:
    df2.to_excel(writer, sheet_name='grafico 9', index=False)

fig, ax = plt.subplots(dpi = 720)

ax.plot(df2[ano], df2['HHI'], linewidth = 0.8),

ax.set(xlabel = 'Year',
       ylabel = 'HHI',
       xlim = (1997,2021))
```

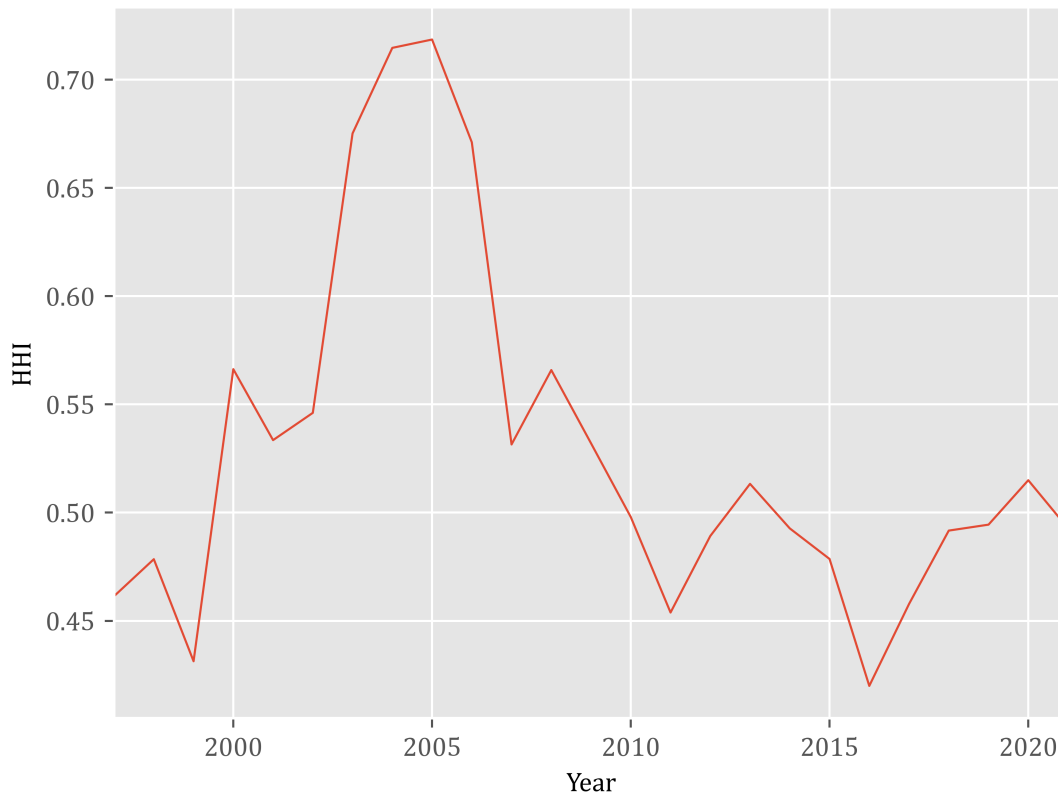
C:\Users\joaop\AppData\Local\Temp\ipykernel\_26652\1657340845.py:8:

FutureWarning: Setting an item of incompatible dtype is deprecated and will

raise an error in a future version of pandas. Value '0.46139887937743196' has dtype incompatible with int64, please explicitly cast to a compatible dtype first.

```
df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i, valor])
```

```
[24]: [Text(0.5, 0, 'Year'), Text(0, 0.5, 'HHI'), (1997.0, 2021.0)]
```



## 11 HHI SH2

```
[25]: SH2 = 'Código SH2'

df1 = df[[SH2, valor, ano]].groupby([SH2, ano], as_index = False).sum()

df1
```

```
[25]:
```

	Código SH2	Ano	Valor FOB (US\$)
0	1	1997	55146
1	1	1998	151138
2	1	1999	106570
3	1	2000	124000
4	1	2001	214400

...	...	...	...
1752	99	2010	232887
1753	99	2012	77913
1754	99	2014	37708
1755	99	2015	197176
1756	99	2016	192900

[1757 rows x 3 columns]

```
[26]: df2 = df1[['ano', 'valor']].groupby(ano, as_index = False).sum()

df2['HHI'] = 0

for i in df2[ano]:

    df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i, 'valor'])

with pd.ExcelWriter(caminho_arquivo, engine='openpyxl', mode='a',
                    if_sheet_exists='replace') as writer:
    df2.to_excel(writer, sheet_name='grafico 10', index=False)

fig, ax = plt.subplots(dpi = 720)

ax.plot(df2[ano], df2['HHI'], linewidth = 0.8),

ax.set(xlabel = 'Year',
       ylabel = 'HHI',
       xlim = (1997, 2021))
```

C:\Users\joaop\AppData\Local\Temp\ipykernel\_26652\531936676.py:7: FutureWarning: Setting an item of incompatible dtype is deprecated and will raise an error in a future version of pandas. Value '0.1823149151026183' has dtype incompatible with int64, please explicitly cast to a compatible dtype first.

```
df2.loc[df2[ano] == i, 'HHI'] = HHI(df1.loc[df1[ano] == i, 'valor'])
```

```
[26]: [Text(0.5, 0, 'Year'), Text(0, 0.5, 'HHI'), (1997.0, 2021.0)]
```

