

```
In [2]: import pandas as pd                    #Importando as bibliotecas que usarei ou pod
import matplotlib.pyplot as plot
import os
import numpy as np
```

```
In [3]: os.chdir(r'C:\Aprendizado\AnaliseDados_Pratica')    #Apontando o caminho dos meus c
```

```
In [276... Tab_Geral = pd.read_csv ('EXP_COMPLETA.csv', sep=";", encoding = "latin1") #Carre
```

```
In [277... Tab_Geral    # Lendo os meus dados
```

```
Out[277]:
```

	CO_ANO	CO_MES	CO_NCM	CO_UNID	CO_PAIS	SG_UF_NCM	CO_VIA	CO_URF	QT_E
0	1997	11	74122000	10	249	SP	4	817600	
1	1997	11	73082000	10	97	MG	7	130151	
2	1997	11	84248119	11	196	SP	1	817800	
3	1997	11	48239090	10	586	MN	9	910600	
4	1997	11	72112910	10	850	SP	1	817800	
...	...	...	...	...	...	...	...	...	...
25878519	2022	10	29032200	10	845	SP	7	1017701	
25878520	2022	9	40169300	10	647	SP	7	1017701	
25878521	2022	12	90184100	11	249	MG	4	817700	
25878522	2022	10	7041000	10	351	SC	1	927700	
25878523	2022	7	7089000	10	301	PA	1	217800	

25878524 rows × 11 columns

```
In [357... Tab_Geral.info()    #Boa prática ao se analisar dados é conferir os tipos de dados c
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25878524 entries, 0 to 25878523
Data columns (total 11 columns):
#   Column      Dtype
---  -
0   CO_ANO      int64
1   CO_MES      int64
2   CO_NCM      int64
3   CO_UNID     int64
4   CO_PAIS     int64
5   SG_UF_NCM   object
6   CO_VIA      int64
7   CO_URF      int64
8   QT_ESTAT   int64
9   KG_LIQUIDO  int64
10  VL_FOB      int64
dtypes: int64(10), object(1)
memory usage: 2.1+ GB
```

```
In [358... Tab_Geral.isnull().sum()    #Fazendo contagem de eventuais dados ausentes
```

```
Out[358]: CO_ANO      0
          CO_MES      0
          CO_NCM      0
          CO_UNID      0
          CO_PAIS      0
          SG_UF_NCM    0
          CO_VIA      0
          CO_URF      0
          QT_ESTAT     0
          KG_LIQUIDO   0
          VL_FOB       0
          dtype: int64
```

```
In [363... if Tab_Geral.duplicated().any(): #Função para verificação de Linhas duplicadas
            print('Existem linhas duplicadas na tabela')
        else:
            print('Não existem linhas duplicadas na tabela')
```

Não existem linhas duplicadas na tabela

```
In [364... Tab_Geral_selecao = Tab_Geral[['CO_ANO','CO_MES','VL_FOB']] #Selecionando as colunas
```

```
In [365... Tab_Geral_selecao
```

Out[365]:

	CO_ANO	CO_MES	VL_FOB
0	1997	11	3
1	1997	11	6442
2	1997	11	31510
3	1997	11	1
4	1997	11	103866
...	...	...	...
25878519	2022	10	470
25878520	2022	9	117
25878521	2022	12	4316
25878522	2022	10	40
25878523	2022	7	47

25878524 rows × 3 columns

```
In [366... Tab_Geral_Agrupado = Tab_Geral_selecao.groupby(['CO_ANO','CO_MES']).sum() #agrupando
```

```
In [367... Tab_Geral_Agrupado
```

Out[367]:

		VL_FOB
CO_ANO	CO_MES	
1997	1	3680971118
	2	3142779857
	3	3824075802
	4	4624157101
	5	4654892460
...	...	...
2022	8	30781272695
	9	28626954294
	10	26868169983
	11	27636417588
	12	26645287802

312 rows × 1 columns

```
In [368... Tab_Geral_Agrupado = Tab_Geral_Agrupado.sort_values(by=['CO_ANO', 'VL_FOB'], ascending=False)
Tab_Geral_Agrupado #Ordenando os valores do maior para o menor
```

Out[368]:

		VL_FOB
CO_ANO	CO_MES	
2022	6	32743972401
	8	30781272695
	7	29845679696
	5	29648653290
	3	29409587271
...	...	...
1997	12	4528699520
	11	3972863432
	3	3824075802
	1	3680971118
	2	3142779857

312 rows × 1 columns

```
In [369... Maior_VLFOB_MES_ANO = Tab_Geral_Agrupado.groupby(['CO_ANO', 'CO_MES'])['VL_FOB'].max()
Maior_VLFOB_MES_ANO #MÊS com maior valor de exportação para cada ano
```

```
In [370... Maior_VLFOB_MES_ANO = Maior_VLFOB_MES_ANO.sort_values(['CO_ANO', 'VL_FOB'], ascending=False)
Maior_VLFOB_MES_ANO #Pegando apenas o primeiro mês com o máximo valor para cada ano , afim de verificar
```

```
In [371... Maior_VLFOB_MES_ANO = Maior_VLFOB_MES_ANO.reset_index()
Maior_VLFOB_MES_ANO
```

```
Out[371]:
```

	CO_ANO	CO_MES
0	1997	7
1	1998	7
2	1999	12
3	2000	8
4	2001	8
5	2002	9
6	2003	10
7	2004	6
8	2005	8
9	2006	8
10	2007	10
11	2008	7
12	2009	12
13	2010	12
14	2011	8
15	2012	5
16	2013	5
17	2014	7
18	2015	6
19	2016	8
20	2017	3
21	2018	10
22	2019	5
23	2020	7
24	2021	6
25	2022	6

```
In [372... Mes_Moda = Maior_VLFOB_MES_ANO['CO_MES'].value_counts().idxmax()
Mes_Moda

#Agosto é o mês que mais se exporta, visto que é o valor que mais se repete na tabela
```

```
Out[372]: 8
```

```
In [373... Tab_Geral_Agrupado.query('CO_ANO == 2002') #Consulta de verificação usando filtro
```

Out[373]:

VL\_FOB

CO\_ANO CO\_MES

CO_ANO	CO_MES	
2002	9	6478828752
	10	6456465000
	7	6202182481
	8	5738521055
	12	5211040751
	11	5102634927
	4	4612069084
	5	4424075300
	3	4251755321
	6	4064856073
	1	3952038148
	2	3652691211

In [374... os.listdir()## Trazer tabela PAÍS para fazer um merge com Tab\_Geral e fazer gráfico

Out[374]:

```
[ 'Analise_1997_2022_Export.ipynb',
  'BckupJupyter',
  'Comandos PANDAS_usados.docx',
  'COMEX - ORIENTACAO.ipynb',
  'Conferencia.twb',
  'EXP_2022.csv',
  'EXP_COMPLETA (1).zip',
  'EXP_COMPLETA.csv',
  'EXP_COMPLETA_MUN.zip',
  'Fonte dos dados é duvidas sobre a base de dados.docx',
  'NCM_SH.csv',
  'PAIS.csv',
  'PAIS_BLOCO.csv',
  'Pandas_comandos.docx',
  'UF.csv',
  'VIA.csv',
  '~Conferencia__12836.twbr']
```

In [375... Tb\_Aux\_Pais = pd.read\_csv(r'PAIS.csv', sep=';', encoding = 'Latin1')

In [376... Tb\_Aux\_Pais

Out[376]:

	CO_PAIS	CO_PAIS_ISON3	CO_PAIS_ISO3	NO_PAIS	NO_PAIS_ING	NO_PAIS_ESP
0	0	898	ZZZ	Não Definido	Not defined	No definido
1	13	4	AFG	Afeganistão	Afghanistan	Afganistan
2	15	248	ALA	Aland, Ilhas	Aland Islands	Alans, Islas
3	17	8	ALB	Albânia	Albania	Albania
4	20	724	ESP	Alboran-Perejil, Ilhas	Alboran-Perejil, Islands	Alboran-Perejil, Islas
...	...	...	...	...	...	...
276	994	898	ZZZ	A Designar	To define	A designar
277	995	898	ZZZ	Bancos Centrais	Central Banks	Bancos Centrales
278	997	898	ZZZ	Organizações Internacionais	International Organizations	Organizaciones Internacionales
279	998	898	ZZZ	Sem informação	Sem informação	Sem informação
280	999	898	ZZZ	Não Declarados	Not declared	No declarados

281 rows × 6 columns

In [377...

Tab\_Geral

Out[377]:

	CO_ANO	CO_MES	CO_NCM	CO_UNID	CO_PAIS	SG_UF_NCM	CO_VIA	CO_URF	QT_E
0	1997	11	74122000	10	249	SP	4	817600	
1	1997	11	73082000	10	97	MG	7	130151	
2	1997	11	84248119	11	196	SP	1	817800	
3	1997	11	48239090	10	586	MN	9	910600	
4	1997	11	72112910	10	850	SP	1	817800	
...	...	...	...	...	...	...	...	...	...
25878519	2022	10	29032200	10	845	SP	7	1017701	
25878520	2022	9	40169300	10	647	SP	7	1017701	
25878521	2022	12	90184100	11	249	MG	4	817700	
25878522	2022	10	7041000	10	351	SC	1	927700	
25878523	2022	7	7089000	10	301	PA	1	217800	

25878524 rows × 11 columns

In [378...

```
Tab_Geral_Pais = pd.merge(Tab_Geral[['CO_PAIS','CO_ANO','CO_MES','SG_UF_NCM','VL_FO
                                , Tb_Aux_Pais[['CO_PAIS','NO_PAIS']], on='CO_PAIS')

#Selecionado colunas da Tabela geral e coluna com o nome do País da tabela auxiliar
```

In [379...

Tab\_Geral\_Pais

Out[379]:

	CO_PAIS	CO_ANO	CO_MES	SG_UF_NCM	VL_FOB	NO_PAIS
0	249	1997	11	SP	3	Estados Unidos
1	249	1997	11	PA	18816	Estados Unidos
2	249	1997	11	RS	22500	Estados Unidos
3	249	1997	11	SC	7120	Estados Unidos
4	249	1997	11	MN	1253	Estados Unidos
...	...	...	...	...	...	...
25878519	343	2020	6	RS	8312	Heard e ilhas mcdonald, Ilha
25878520	343	2022	8	ES	245	Heard e ilhas mcdonald, Ilha
25878521	343	2022	8	RJ	302091	Heard e ilhas mcdonald, Ilha
25878522	343	2022	6	RS	1378	Heard e ilhas mcdonald, Ilha
25878523	292	2022	7	SP	71526	Geórgia do Sul e Sandwich do Sul, Ilhas

25878524 rows × 6 columns

In [380... Tabela\_Agrupada\_Geral\_Pais = Tab\_Geral\_Pais.groupby('NO\_PAIS').sum('VL\_FOB').sort\_

In [381... Top\_10 = Tabela\_Agrupada\_Geral\_Pais.head(10)  
Top\_10

Out[381]:

	CO_PAIS	CO_ANO	CO_MES	VL_FOB
NO_PAIS				
China	56834720	715001876	2366699	775048087479
Estados Unidos	444802893	3592529601	11845640	569233641353
Argentina	144225774	4601402195	15164154	308184044293
Países Baixos (Holanda)	153307869	538165143	1768256	187480835451
Alemanha	13262306	1159491178	3787216	133800813699
Japão	120404235	606932100	1989031	121118644326
Chile	218468970	2780344597	9195834	99213062273
México	455831251	1859590427	6125861	91945798633
Itália	143931680	749582070	2424280	86612543110
Espanha	67361280	552692437	1798350	81191432816

In [382... Top\_10 = Top\_10.reset\_index() #resetando para fazer a plotagem

In [383... Top\_10 = Top\_10.sort\_values(by='VL\_FOB', ascending=True) #ordeando para plotar de  
Top\_10

Out[383]:

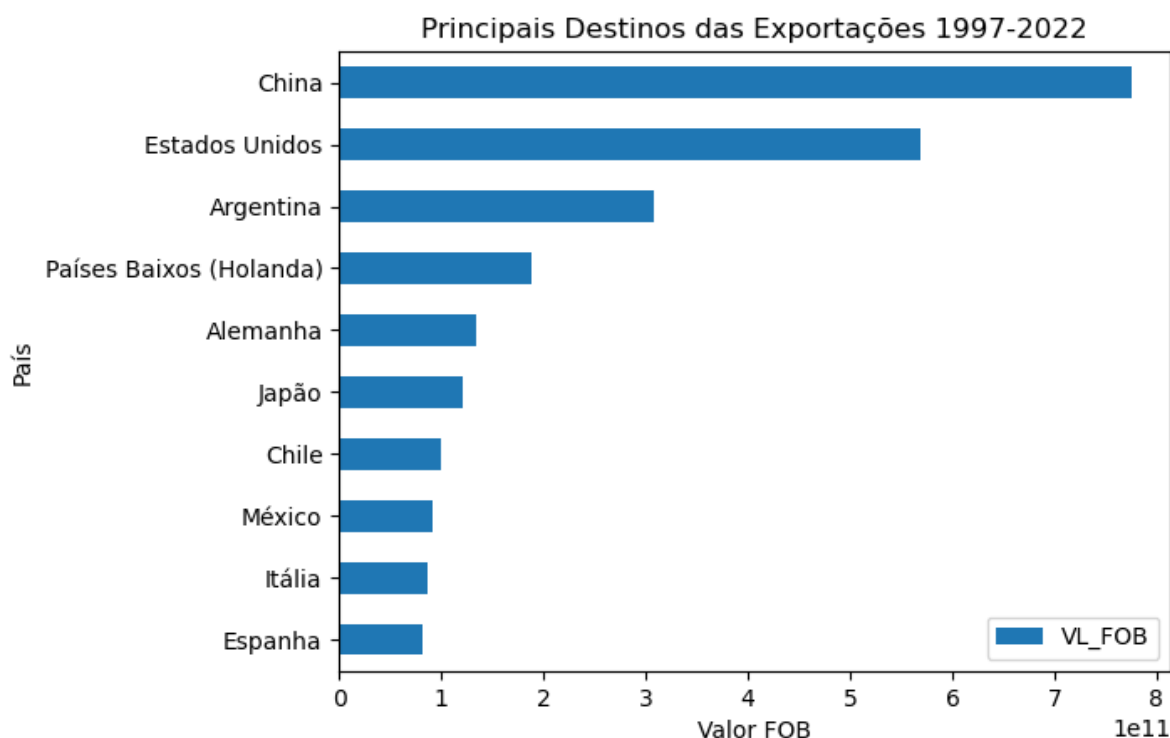
	NO_PAIS	CO_PAIS	CO_ANO	CO_MES	VL_FOB
9	Espanha	67361280	552692437	1798350	81191432816
8	Itália	143931680	749582070	2424280	86612543110
7	México	455831251	1859590427	6125861	91945798633
6	Chile	218468970	2780344597	9195834	99213062273
5	Japão	120404235	606932100	1989031	121118644326
4	Alemanha	13262306	1159491178	3787216	133800813699
3	Países Baixos (Holanda)	153307869	538165143	1768256	187480835451
2	Argentina	144225774	4601402195	15164154	308184044293
1	Estados Unidos	444802893	3592529601	11845640	569233641353
0	China	56834720	715001876	2366699	775048087479

In [384...

```
# gráfico de barras horizontais
Top_10.plot(x='NO_PAIS', y='VL_FOB', kind='barh', legend=True)

# título do gráfico e dos eixos
plot.title('Principais Destinos das Exportações 1997-2022')
plot.xlabel('Valor FOB')
plot.ylabel('País')

# Exibir
plot.show()
```



In [385...

```
Tab_Geral_Pais_UF = Tab_Geral_Pais.groupby('SG_UF_NCM').sum('VL_FOB') #agrupando e
Tab_Geral_Pais_UF = Tab_Geral_Pais_UF.sort_values('VL_FOB', ascending=False) #ordenando
Tab_Geral_Pais_UF = Tab_Geral_Pais_UF.head(10) #top 10 dos estados com maior valor
Tab_Geral_Pais_UF
```



Out[385]:

	CO_PAIS	CO_ANO	CO_MES	VL_FOB
SG_UF_NCM				
SP	3859141173	21128465626	69889365	1087090663004
MG	567709206	3158569838	10461127	540623092007
RJ	390434410	2166687516	7191619	404058869855
RS	1419290990	7128481525	23699775	358023122739
PR	752424857	3937436599	13095946	326472671096
PA	78887700	411382964	1366996	274794632602
MT	41974521	237497408	790137	251948691477
ES	138760210	711843317	2395904	179979039504
BA	149832532	839398035	2786769	175666267597
SC	861986502	4427706270	14862383	174460986003

In [386...

```
Tab_Geral_Pais_UF = Tab_Geral_Pais_UF.reset_index() #resetando o index para poder
Tab_Geral_Pais_UF
```

Out[386]:

	SG_UF_NCM	CO_PAIS	CO_ANO	CO_MES	VL_FOB
0	SP	3859141173	21128465626	69889365	1087090663004
1	MG	567709206	3158569838	10461127	540623092007
2	RJ	390434410	2166687516	7191619	404058869855
3	RS	1419290990	7128481525	23699775	358023122739
4	PR	752424857	3937436599	13095946	326472671096
5	PA	78887700	411382964	1366996	274794632602
6	MT	41974521	237497408	790137	251948691477
7	ES	138760210	711843317	2395904	179979039504
8	BA	149832532	839398035	2786769	175666267597
9	SC	861986502	4427706270	14862383	174460986003

In [387...

```
# Cria o gráfico de barras horizontais
Tab_Geral_Pais_UF.plot(x='SG_UF_NCM', y='VL_FOB',kind='bar', legend=True)

# Define o título do gráfico e dos eixos
plot.title('Exportações por Estado 1997-2022')
plot.xlabel('Estado')
plot.ylabel('Valor FOB')

# Exibe o gráfico
plot.show()
```



In [388...

```
Tab_Geral_ANO = Tab_Geral.groupby('CO_ANO').sum('VL_FOB')  
Tab_Geral_ANO = Tab_Geral_ANO.drop(columns=['CO_MES', 'CO_NCM', 'CO_VIA', 'CO_UNID', 'CO'])  
Tab_Geral_ANO
```

Out[388]:

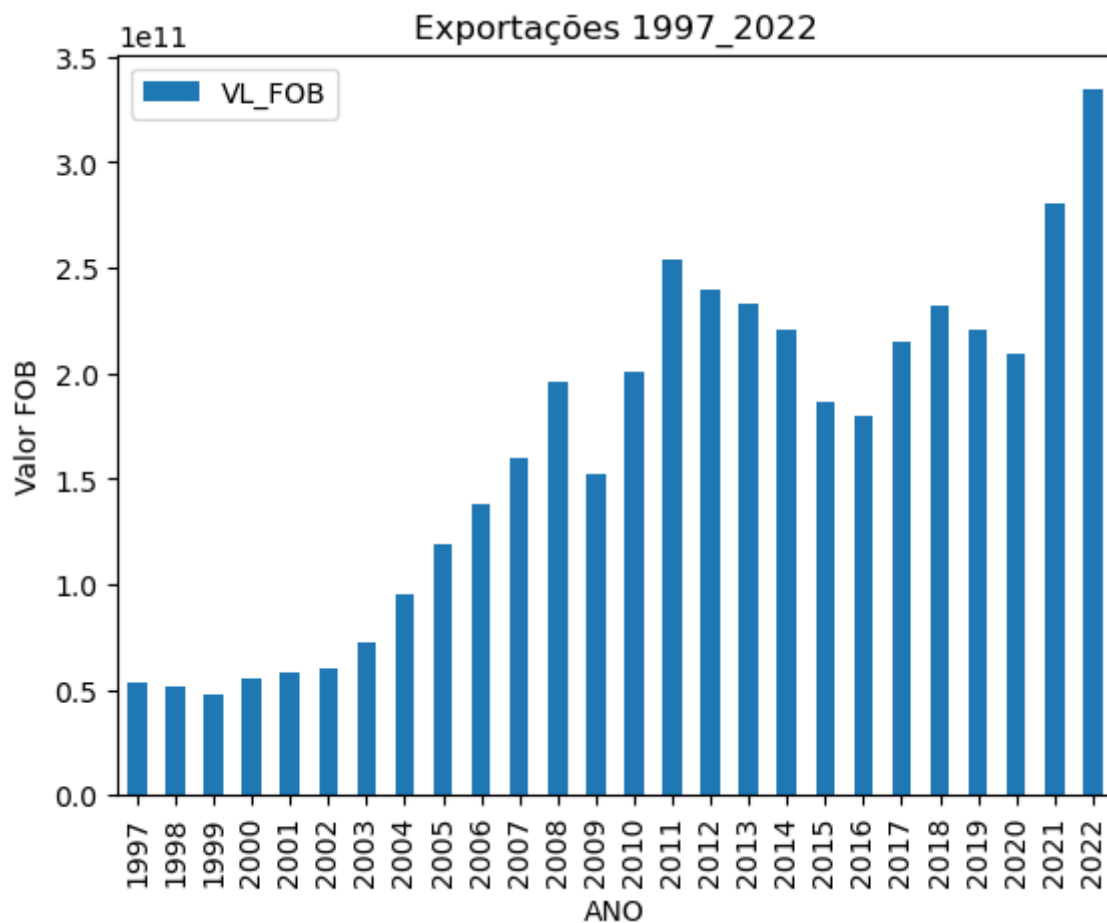
VL_FOB	
CO_ANO	
1997	52947495532
1998	51076603549
1999	47945909310
2000	54993159648
2001	58032294243
2002	60147158103
2003	72776746690
2004	95121672369
2005	118597835407
2006	137581151209
2007	159816383833
2008	195764624177
2009	151791674186
2010	200434134826
2011	253666309507
2012	239952538158
2013	232544255606
2014	220923236838
2015	186782355063
2016	179526129214
2017	214988108353
2018	231889523399
2019	221126807647
2020	209180241655
2021	280814577460
2022	334463079195

```
In [389...] Tab_Geral_ANO = Tab_Geral_ANO.reset_index() #Resetando index para plotar

In [390...] #Grafico de barras verticais para apresentar a evolução ano a ano das exportações e
# Cria o gráfico de barras horizontais
Tab_Geral_ANO.plot(x='CO_ANO', y='VL_FOB',kind='bar', legend=True)

# Define o título do gráfico e dos eixos
plot.title('Exportações 1997_2022')
plot.xlabel('ANO')
plot.ylabel('Valor FOB')
```

```
# Exibe o gráfico  
plot.show()
```



In [391...

```
tabela_BL_Economico = pd.read_csv(r'PAIS_BLOCO.csv', sep=';', encoding = 'Latin1')  
tabela_BL_Economico #Carregando minha tabela auxiliar com o nome bloco econômico
```

Out[391]:

	CO_PAIS	CO_BLOCO	NO_BLOCO	NO_BLOCO_ING	NO_BLOCO_ESP
0	788	51	África	Africa (minus MIDDLE EAST)	África (menos ORIENTE MEDIO)
1	525	51	África	Africa (minus MIDDLE EAST)	África (menos ORIENTE MEDIO)
2	229	51	África	Africa (minus MIDDLE EAST)	África (menos ORIENTE MEDIO)
3	281	51	África	Africa (minus MIDDLE EAST)	África (menos ORIENTE MEDIO)
4	438	51	África	Africa (minus MIDDLE EAST)	África (menos ORIENTE MEDIO)
...	...	...	...	...	...
317	321	22	União Europeia - UE	European Union (EU)	Uniao Europea (UE)
318	271	22	União Europeia - UE	European Union (EU)	Uniao Europea (UE)
319	251	22	União Europeia - UE	European Union (EU)	Uniao Europea (UE)
320	607	22	União Europeia - UE	European Union (EU)	Uniao Europea (UE)
321	23	22	União Europeia - UE	European Union (EU)	Uniao Europea (UE)

322 rows × 5 columns

In [392...]

```
Tab_Geral_Pais_Bloco = pd.merge (Tab_Geral_Pais[['CO_PAIS', 'CO_ANO', 'CO_MES', 'SG_UI
                                tabela_BL_Economico[['CO_PAIS', 'NO_BLOCO']], on='C

#Unindo a coluna nome do Bloco economico da tabela auxiliar na minha tabela princip
```

In [393...]

```
Tab_Geral_Pais_Bloco
```

Out[393]:

	CO_PAIS	CO_ANO	CO_MES	SG_UF_NCM	VL_FOB	NO_PAIS	NO_BLOCO
0	249	1997	11	SP	3	Estados Unidos	América do Norte
1	249	1997	11	PA	18816	Estados Unidos	América do Norte
2	249	1997	11	RS	22500	Estados Unidos	América do Norte
3	249	1997	11	SC	7120	Estados Unidos	América do Norte
4	249	1997	11	MN	1253	Estados Unidos	América do Norte
...	...	...	...	...	...	...	...
35325382	321	2022	2	SP	155	Guernsey	União Europeia - UE
35325383	321	2022	6	SP	385	Guernsey	Europa
35325384	321	2022	6	SP	385	Guernsey	União Europeia - UE
35325385	321	2022	8	SP	17	Guernsey	Europa
35325386	321	2022	8	SP	17	Guernsey	União Europeia - UE

35325387 rows × 7 columns

In [394...  
Tab\_Bloco = Tab\_Geral\_Pais\_Bloco.groupby('NO\_BLOCO').sum('VL\_FOB') #Fazendo agregação  
Tab\_Bloco = Tab\_Bloco.sort\_values('VL\_FOB',ascending=False)

In [395...  
Tab\_Bloco

Out[395]:

	CO_PAIS	CO_ANO	CO_MES	VL_FOB
NO_BLOCO				
Ásia (Exclusive Oriente Médio)	822873752	4151483644	13760886	1324588404837
Europa	1576112587	8249001661	27020071	967411638679
União Europeia - UE	1033816562	6594326509	21570219	749215619598
América do Norte	934156462	5904476964	19456292	719322432914
América do Sul	4081552975	22321177129	74098497	694224407872
Mercado Comum do Sul - Mercosul	2432895278	11186924332	37130688	401366995998
Oriente Médio	171759622	1246588578	4118098	194743409416
Associação de Nações do Sudeste Asiático - ASEAN	401160728	1294610499	4315317	184523791809
África	718064720	3857115887	12890516	183637595627
América Central e Caribe	1106934475	5250678656	17619298	111635834331
Oceania	156110798	994540779	3340152	15616986833

```
In [396... Tab_Bloco = Tab_Bloco.drop(columns=['CO_PAIS', 'CO_ANO', 'CO_MES']) #Exluindo colunas

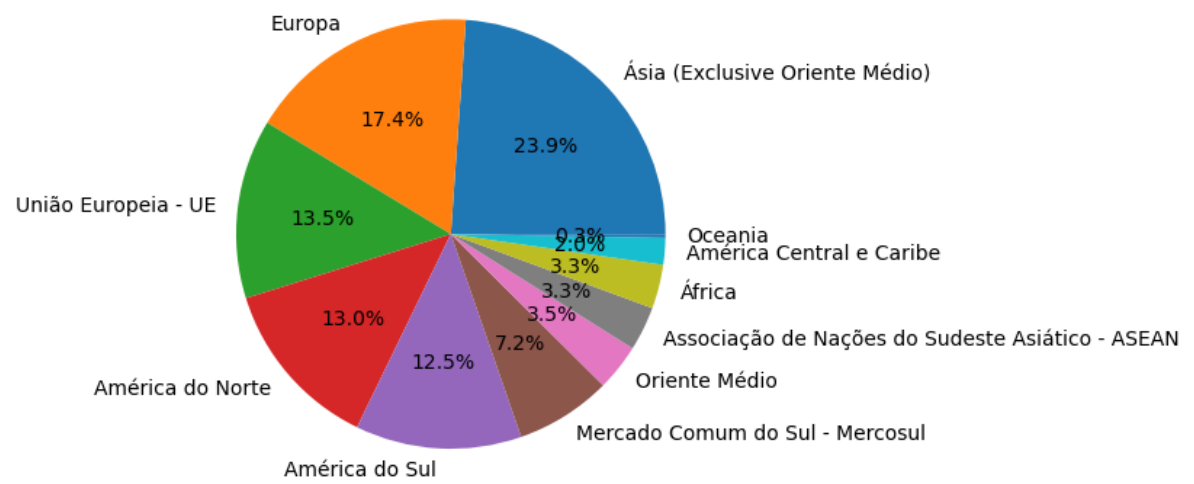
In [397... Tab_Bloco

Out[397]:
```

VL_FOB	
NO_BLOCO	
Ásia (Exclusive Oriente Médio)	1324588404837
Europa	967411638679
União Europeia - UE	749215619598
América do Norte	719322432914
América do Sul	694224407872
Mercado Comum do Sul - Mercosul	401366995998
Oriente Médio	194743409416
Associação de Nações do Sudeste Asiático - ASEAN	184523791809
África	183637595627
América Central e Caribe	111635834331
Oceania	15616986833

```
In [398... Tab_Bloco = Tab_Bloco.reset_index() #Faço o reset index para plotar o gráfico

In [399... #Crie um gráfico de pizza com rótulos reduzidos
fig, ax = plot.subplots()
ax.pie(Tab_Bloco['VL_FOB'], labels=Tab_Bloco['NO_BLOCO'], autopct='%1.1f%%', textpct='%1.1f%%')
# Mostre o gráfico
plot.show()
```



```
In [ ]: #Conforme gráfico acima vemos que temos O continente Europeu como bloco econômico o
#seguido da Ásia.

In [ ]:
```