

APS4

November 18, 2022

1 PARTE 3

1.1 Introdução

O objetivo da APS3 é realizar uma análise inferencial com base nas probabilidades calculadas pelo grupo. Para isso, escolhemos dois blocos econômicos para trabalharmos: O Mercosul e a União Europeia.

```
[3]: import pandas as pd
```

1.2 Tratamento e Importação dos dados

```
[4]: PATH = 'data/API_EN.ATM.CO2E.PC_DS2_en_csv_v2_4353266.csv'
df = pd.read_csv(PATH, skiprows=4, sep=',', decimal='.', encoding='latin1')

# Utilizar apenas os dados disponíveis (entre 1960 e 2019)
for i in range(1960, 2000):
    del df[str(i)]
df.drop(['2020', '2021', 'Unnamed: 66'], axis=1, inplace=True)

df.head(4)
```

```
[4]:
```

	Country Name	Country Code	\
0	Aruba	ABW	
1	Africa Eastern and Southern	AFE	
2	Afghanistan	AFG	
3	Africa Western and Central	AFW	

	Indicator Name	Indicator Code	2000	2001	\
0	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	NaN	NaN	
1	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	0.894017	0.962371	
2	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	0.036574	0.033785	
3	CO2 emissions (metric tons per capita)	EN.ATM.CO2E.PC	0.530435	0.544612	

	2002	2003	2004	2005	...	2010	2011	2012	\
0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	
1	0.964187	0.991013	1.036161	1.011151	...	1.021548	0.979352	0.992616	
2	0.045574	0.051518	0.041655	0.060419	...	0.243614	0.296506	0.259295	

```
3 0.502621 0.521609 0.507780 0.508247 ... 0.472077 0.476789 0.480603
```

```

      2013      2014      2015      2016      2017      2018      2019
0      NaN      NaN      NaN      NaN      NaN      NaN      NaN
1  1.005027  1.016649  0.966589  0.948410  0.937926  0.917507  0.913618
2  0.185624  0.146236  0.172897  0.149789  0.131695  0.163295  0.159824
3  0.508099  0.515960  0.494065  0.499405  0.483140  0.486456  0.493923

```

[4 rows x 24 columns]

```
[5]: # Filtragem
df_ue = df[df['Country Code'].isin(['AUT', 'BEL', 'BGR', 'HRV', 'CYP', 'CZE',
    ↪ 'DNK', 'EST', 'FIN', 'FRA', 'DEU', 'GRC', 'HUN', 'IRL', 'ITA', 'LVA', 'LTU',
    ↪ 'LUX', 'MLT', 'NLD', 'POL', 'PRT', 'ROU', 'SVK', 'SVN', 'ESP', 'SWE'])]

df_ms = df[df['Country Code'].isin(['ARG', 'BRA', 'PRY', 'URY', 'VEN', 'CHL',
    ↪ 'COL', 'PER', 'BOL', 'ECU', 'GUY', 'SUR'])]
```

```
[6]: df_ue.head(4)
```

```
[6]: Country Name Country Code Indicator Name \
14 Austria AUT CO2 emissions (metric tons per capita)
17 Belgium BEL CO2 emissions (metric tons per capita)
21 Bulgaria BGR CO2 emissions (metric tons per capita)
53 Cyprus CYP CO2 emissions (metric tons per capita)

Indicator Code 2000 2001 2002 2003 2004 \
14 EN.ATM.CO2E.PC 7.929786 8.444109 8.583317 9.168344 9.275614
17 EN.ATM.CO2E.PC 11.439581 11.504321 10.727988 11.131315 10.941224
21 EN.ATM.CO2E.PC 5.314454 5.767160 5.563239 6.164371 6.125549
53 EN.ATM.CO2E.PC 7.558667 7.279172 7.339033 7.880736 7.699845

2005 ... 2010 2011 2012 2013 2014 \
14 9.266114 ... 8.365015 8.135474 7.723614 7.753700 7.260404
17 10.555782 ... 9.794792 8.740505 8.577526 8.655478 8.041711
21 6.265854 ... 6.049544 6.755278 6.162153 5.458964 5.820925
53 7.687390 ... 7.100377 6.783205 6.255253 5.621288 5.909935

2015 2016 2017 2018 2019
14 7.317159 7.288820 7.486162 7.133064 7.293984
17 8.437852 8.314932 8.151096 8.196339 8.095584
21 6.207865 5.834882 6.201290 5.822034 5.610857
53 5.900152 6.169943 6.238954 6.054175 5.998795

```

[4 rows x 24 columns]

```
[7]: df_ms.head(4)
```

```
[7]: Country Name Country Code Indicator Name \
9 Argentina ARG CO2 emissions (metric tons per capita)
28 Bolivia BOL CO2 emissions (metric tons per capita)
29 Brazil BRA CO2 emissions (metric tons per capita)
39 Chile CHL CO2 emissions (metric tons per capita)

Indicator Code 2000 2001 2002 2003 2004 \
9 EN.ATM.CO2E.PC 3.587392 3.360371 3.117425 3.351724 3.672974
28 EN.ATM.CO2E.PC 0.975260 0.938202 0.949350 1.019558 1.046417
29 EN.ATM.CO2E.PC 1.794550 1.802410 1.769881 1.709539 1.785372
39 EN.ATM.CO2E.PC 3.286328 3.121272 3.131772 3.167263 3.432413

2005 ... 2010 2011 2012 2013 2014 2015 \
9 3.753639 ... 4.099690 4.280989 4.264224 4.342115 4.209096 4.301914
28 1.088569 ... 1.509663 1.626366 1.678603 1.777588 1.906316 1.909891
29 1.782062 ... 2.033226 2.117869 2.279272 2.421657 2.523232 2.373629
39 3.458629 ... 4.087905 4.437266 4.506229 4.723555 4.311061 4.576124

2016 2017 2018 2019
9 4.201846 4.071308 3.975772 3.740650
28 1.995137 2.032547 2.046130 1.940398
29 2.168575 2.196418 2.071855 2.057811
39 4.749830 4.714020 4.624338 4.821118
```

[4 rows x 24 columns]

2 Sobre os blocos econômicos escolhidos

2.1 Mercosul

O Mercosul é uma iniciativa de livre comércio de países sul-americanos. Conta, atualmente, com 5 países membros e 7 países associados. O bloco foi fundado em 1991 e, desde então, tem como objetivo a integração econômica e social dos países que o compõem. O maior critério para a criação e adesão de países é a proximidade geográfica e cultural de seus membros.

2.2 União Europeia

A União Europeia é uma iniciativa que institui um mercado comum no continente europeu. O bloco foi fundado em 1993 e atualmente conta com 27 países membros. O objetivo da União Europeia é a integração econômica e social dos países que a compõem. Os critérios de entrada e permanência no bloco são mais complexos que os do Mercosul, envolvendo os critérios de Copenhague (questões políticas, econômicas e de acervo comunitário). Além do mercado comum, o bloco também possui uma política externa comum, uma moeda comum (Euro) e uma política de imigração comum.

```
[8]: # Países da União Europeia
df_ue['Country Name'].unique()
```

```
[8]: array(['Austria', 'Belgium', 'Bulgaria', 'Cyprus', 'Czech Republic',
        'Germany', 'Denmark', 'Spain', 'Estonia', 'Finland', 'France',
        'Greece', 'Croatia', 'Hungary', 'Ireland', 'Italy', 'Lithuania',
        'Luxembourg', 'Latvia', 'Malta', 'Netherlands', 'Poland',
        'Portugal', 'Romania', 'Slovak Republic', 'Slovenia', 'Sweden'],
        dtype=object)
```

```
[9]: # Países do Mercosul
df_ms['Country Name'].unique()
```

```
[9]: array(['Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia', 'Ecuador',
        'Guyana', 'Peru', 'Paraguay', 'Suriname', 'Uruguay',
        'Venezuela, RB'], dtype=object)
```

3 Métrica escolhida

Para a análise, o grupo escolheu trabalhar com o Delta de emissões de CO2 por pessoa entre 2000 e 2019. O Delta foi escolhido pois demonstra mais facilmente as variações de emissões de CO2 por pessoa entre os países, podendo ser facilmente comparado e analisado. Para calculá-lo, o grupo realizou a contagem de emissões de CO2 por pessoa em 2000 e em 2019, e então subtraiu o valor de 2000 do valor de 2019. O resultado foi o Delta apresentado a seguir.

Para o Status, o grupo utilizou a seguinte classificação: - Delta maior ou igual a zero: Status Atrasado - Delta menor que zero: Status Avançado

```
[10]: # Calcular o Delta de emissões de CO2 por país de 2000 até 2019
for country in df_ue['Country Name'].unique():
    df_ue.loc[df_ue['Country Name'] == country, 'Delta'] = df_ue.
    ↪loc[df_ue['Country Name'] == country, '2019'].values[0] - df_ue.
    ↪loc[df_ue['Country Name'] == country, '2000'].values[0]

for country in df_ms['Country Name'].unique():
    df_ms.loc[df_ms['Country Name'] == country, 'Delta'] = df_ms.
    ↪loc[df_ms['Country Name'] == country, '2019'].values[0] - df_ms.
    ↪loc[df_ms['Country Name'] == country, '2000'].values[0]

df_ue.head(4)
```

/tmp/ipykernel_6686/3508682445.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ue.loc[df_ue['Country Name'] == country, 'Delta'] =
df_ue.loc[df_ue['Country Name'] == country, '2019'].values[0] -
df_ue.loc[df_ue['Country Name'] == country, '2000'].values[0]
```

```
/tmp/ipykernel_6686/3508682445.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_ms.loc[df_ms['Country Name'] == country, 'Delta'] =
df_ms.loc[df_ms['Country Name'] == country, '2019'].values[0] -
df_ms.loc[df_ms['Country Name'] == country, '2000'].values[0]
```

```
[10]: Country Name Country Code Indicator Name \
14 Austria AUT CO2 emissions (metric tons per capita)
17 Belgium BEL CO2 emissions (metric tons per capita)
21 Bulgaria BGR CO2 emissions (metric tons per capita)
53 Cyprus CYP CO2 emissions (metric tons per capita)

Indicator Code 2000 2001 2002 2003 2004 \
14 EN.ATM.CO2E.PC 7.929786 8.444109 8.583317 9.168344 9.275614
17 EN.ATM.CO2E.PC 11.439581 11.504321 10.727988 11.131315 10.941224
21 EN.ATM.CO2E.PC 5.314454 5.767160 5.563239 6.164371 6.125549
53 EN.ATM.CO2E.PC 7.558667 7.279172 7.339033 7.880736 7.699845

2005 ... 2011 2012 2013 2014 2015 \
14 9.266114 ... 8.135474 7.723614 7.753700 7.260404 7.317159
17 10.555782 ... 8.740505 8.577526 8.655478 8.041711 8.437852
21 6.265854 ... 6.755278 6.162153 5.458964 5.820925 6.207865
53 7.687390 ... 6.783205 6.255253 5.621288 5.909935 5.900152

2016 2017 2018 2019 Delta
14 7.288820 7.486162 7.133064 7.293984 -0.635801
17 8.314932 8.151096 8.196339 8.095584 -3.343997
21 5.834882 6.201290 5.822034 5.610857 0.296404
53 6.169943 6.238954 6.054175 5.998795 -1.559872
```

[4 rows x 25 columns]

```
[11]: df_ms.head(4)
```

```
[11]: Country Name Country Code Indicator Name \
9 Argentina ARG CO2 emissions (metric tons per capita)
28 Bolivia BOL CO2 emissions (metric tons per capita)
29 Brazil BRA CO2 emissions (metric tons per capita)
39 Chile CHL CO2 emissions (metric tons per capita)

Indicator Code 2000 2001 2002 2003 2004 \
9 EN.ATM.CO2E.PC 3.587392 3.360371 3.117425 3.351724 3.672974
28 EN.ATM.CO2E.PC 0.975260 0.938202 0.949350 1.019558 1.046417
```

```

29  EN.ATM.CO2E.PC  1.794550  1.802410  1.769881  1.709539  1.785372
39  EN.ATM.CO2E.PC  3.286328  3.121272  3.131772  3.167263  3.432413

```

```

      2005  ...      2011      2012      2013      2014      2015      2016  \
9    3.753639  ...  4.280989  4.264224  4.342115  4.209096  4.301914  4.201846
28   1.088569  ...  1.626366  1.678603  1.777588  1.906316  1.909891  1.995137
29   1.782062  ...  2.117869  2.279272  2.421657  2.523232  2.373629  2.168575
39   3.458629  ...  4.437266  4.506229  4.723555  4.311061  4.576124  4.749830

```

```

      2017      2018      2019      Delta
9    4.071308  3.975772  3.740650  0.153258
28   2.032547  2.046130  1.940398  0.965138
29   2.196418  2.071855  2.057811  0.263261
39   4.714020  4.624338  4.821118  1.534790

```

[4 rows x 25 columns]

```

[12]: df_ue['Status'] = df_ue['Delta'].apply(lambda x: 'Atrasado' if x >= 0 else
      ↪ 'Avançado')
      df_ms['Status'] = df_ms['Delta'].apply(lambda x: 'Atrasado' if x >= 0 else
      ↪ 'Avançado')

```

/tmp/ipykernel_6686/3716828161.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```

df_ue['Status'] = df_ue['Delta'].apply(lambda x: 'Atrasado' if x >= 0 else
'Avançado')

```

/tmp/ipykernel_6686/3716828161.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```

df_ms['Status'] = df_ms['Delta'].apply(lambda x: 'Atrasado' if x >= 0 else
'Avançado')

```

```

[13]: # concatenar os dois blocos
df_concat = pd.concat([df_ue, df_ms])
df_concat['Bloco'] = df_concat['Country Code'].apply(lambda x: 'UE' if x in
      ↪ ['AUT', 'BEL', 'BGR', 'HRV', 'CYP', 'CZE', 'DNK', 'EST', 'FIN', 'FRA',
      ↪ 'DEU', 'GRC', 'HUN', 'IRL', 'ITA', 'LVA', 'LTU', 'LUX', 'MLT', 'NLD', 'POL',
      ↪ 'PRT', 'ROU', 'SVK', 'SVN', 'ESP', 'SWE'] else 'Mercosul')
df_concat.head()

```

```
[13]: Country Name Country Code Indicator Name \
14 Austria AUT CO2 emissions (metric tons per capita)
17 Belgium BEL CO2 emissions (metric tons per capita)
21 Bulgaria BGR CO2 emissions (metric tons per capita)
53 Cyprus CYP CO2 emissions (metric tons per capita)
54 Czech Republic CZE CO2 emissions (metric tons per capita)

Indicator Code 2000 2001 2002 2003 2004 \
14 EN.ATM.CO2E.PC 7.929786 8.444109 8.583317 9.168344 9.275614
17 EN.ATM.CO2E.PC 11.439581 11.504321 10.727988 11.131315 10.941224
21 EN.ATM.CO2E.PC 5.314454 5.767160 5.563239 6.164371 6.125549
53 EN.ATM.CO2E.PC 7.558667 7.279172 7.339033 7.880736 7.699845
54 EN.ATM.CO2E.PC 12.010653 12.011818 11.624103 12.043361 12.105402

2005 ... 2013 2014 2015 2016 2017 \
14 9.266114 ... 7.753700 7.260404 7.317159 7.288820 7.486162
17 10.555782 ... 8.655478 8.041711 8.437852 8.314932 8.151096
21 6.265854 ... 5.458964 5.820925 6.207865 5.834882 6.201290
53 7.687390 ... 5.621288 5.909935 5.900152 6.169943 6.238954
54 11.750804 ... 9.620257 9.264303 9.400668 9.627750 9.611647

2018 2019 Delta Status Bloco
14 7.133064 7.293984 -0.635801 Avançado UE
17 8.196339 8.095584 -3.343997 Avançado UE
21 5.822034 5.610857 0.296404 Atrasado UE
53 6.054175 5.998795 -1.559872 Avançado UE
54 9.492068 9.022786 -2.987867 Avançado UE
```

[5 rows x 27 columns]

```
[14]: # Contagem de países por Status
df_concat['Status'].value_counts()
```

```
[14]: Avançado    23
Atrasado      16
Name: Status, dtype: int64
```

```
[15]: # Contagem de países por Bloco e Status
df_concat.groupby(['Bloco', 'Status']).size()
```

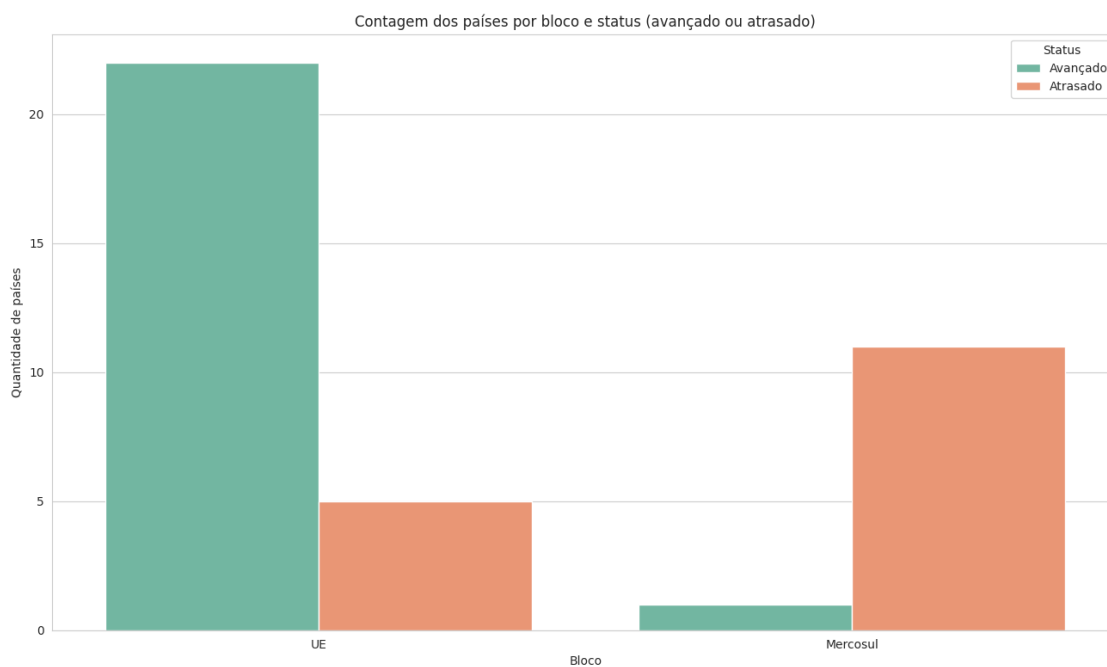
```
[15]: Bloco    Status
Mercosul  Atrasado    11
          Avançado     1
UE        Atrasado     5
          Avançado    22
dtype: int64
```

4 Análise Exploratória

```
[16]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(16, 9))
sns.set_style('whitegrid')
sns.countplot(x='Bloco', hue='Status', data=df_concat, palette='Set2')
plt.title('Contagem dos países por bloco e status (avanzado ou atrasado)')
plt.ylabel('Quantidade de países')
plt.xlabel('Bloco')
```

```
[16]: Text(0.5, 0, 'Bloco')
```



5 Calculando as Probabilidades e Estatísticas

```
[17]: # P(Avanzado | UE)
p_avanzado_ue = df_concat[(df_concat['Bloco'] == 'UE') & (df_concat['Status'] == 'Avanzado')].shape[0] / df_concat[df_concat['Bloco'] == 'UE'].shape[0]
# P(Atrasado | UE)
p_atrasado_ue = df_concat[(df_concat['Bloco'] == 'UE') & (df_concat['Status'] == 'Atrasado')].shape[0] / df_concat[df_concat['Bloco'] == 'UE'].shape[0]
print(f'A probabilidade de um país do bloco UE estar avanzado em relação à métrica é de {p_avanzado_ue*100:.2f}%')
```



```
print(f'A probabilidade de um país do bloco UE estar atrasado em relação à
↳ métrica é de {p_atrasado_ue*100:.2f}%.'
```

A probabilidade de um país do bloco UE estar avançado em relação à métrica é de 81.48%.

A probabilidade de um país do bloco UE estar atrasado em relação à métrica é de 18.52%.

```
[18]: # P(Avançado / Mercosul)
p_avancado_ms = df_concat[(df_concat['Bloco'] == 'Mercosul') &
↳ (df_concat['Status'] == 'Avançado')].shape[0] / df_concat[df_concat['Bloco']
↳ == 'Mercosul'].shape[0]
# P(Atrasado / Mercosul)
p_atrasado_ms = df_concat[(df_concat['Bloco'] == 'Mercosul') &
↳ (df_concat['Status'] == 'Atrasado')].shape[0] / df_concat[df_concat['Bloco']
↳ == 'Mercosul'].shape[0]
print(f'A probabilidade de um país do bloco Mercosul estar avançado em relação
↳ à métrica é de {p_avancado_ms*100:.2f}%.'
```

A probabilidade de um país do bloco Mercosul estar avançado em relação à métrica é de 8.33%.

A probabilidade de um país do bloco Mercosul estar atrasado em relação à métrica é de 91.67%.

```
[19]: # P(Mercosul)
p_ms = df_concat[df_concat['Bloco'] == 'Mercosul'].shape[0] / df_concat.shape[0]
# P(UE)
p_ue = df_concat[df_concat['Bloco'] == 'UE'].shape[0] / df_concat.shape[0]
print(f'A probabilidade de um país escolhido ser do bloco Mercosul é de
↳ {p_ms*100:.2f}%.'
```

A probabilidade de um país escolhido ser do bloco Mercosul é de 30.77%.

A probabilidade de um país escolhido ser do bloco UE é de 69.23%.

```
[20]: # P(Avançado)
p_avancado = df_concat[df_concat['Status'] == 'Avançado'].shape[0] / df_concat.
↳ shape[0]
# P(Atrasado)
p_atrasado = df_concat[df_concat['Status'] == 'Atrasado'].shape[0] / df_concat.
↳ shape[0]
print(f'A probabilidade de um país escolhido estar avançado em relação à
↳ métrica é de {p_avancado*100:.2f}%.'
```

```
print(f'A probabilidade de um país escolhido estar atrasado em relação à
↳ métrica é de {p_atrasado*100:.2f}%.'
```

A probabilidade de um país escolhido estar avançado em relação à métrica é de 58.97%.

A probabilidade de um país escolhido estar atrasado em relação à métrica é de 41.03%.

```
[21]: # P(Mercosul / Avançado) = P(Mercosul) * P(Avançado / Mercosul) / P(Avançado)
p_ms_avancado = (p_avancado_ms * p_ms) / p_avancado
print(f'De acordo com o Bayesiano calculado, a probabilidade de um país ser do
↳ Mercosul dado que ele está avançado em relação à métrica é de
↳ {p_ms_avancado*100:.1f}%.'
```

De acordo com o Bayesiano calculado, a probabilidade de um país ser do Mercosul dado que ele está avançado em relação à métrica é de 4.3%.

```
[22]: # P(Mercosul / Atrasado) = P(Mercosul) * P(Atrasado / Mercosul) / P(Atrasado)
p_ms_atrasado = (p_atrasado_ms * p_ms) / p_atrasado
print(f'De acordo com o Bayesiano calculado, a probabilidade de um país ser do
↳ Mercosul dado que ele está atrasado em relação à métrica é de
↳ {p_ms_atrasado*100:.1f}%.'
```

De acordo com o Bayesiano calculado, a probabilidade de um país ser do Mercosul dado que ele está atrasado em relação à métrica é de 68.8%.

```
[23]: # P(UE / Avançado) = P(UE) * P(Avançado / UE) / P(Avançado)
p_ue_avancado = (p_avancado_ue * p_ue) / p_avancado
print(f'De acordo com o Bayesiano calculado, a probabilidade de um país ser da
↳ UE dado que ele está avançado em relação à métrica é de {p_ue_avancado*100:.
↳ 1f}%.'
```

De acordo com o Bayesiano calculado, a probabilidade de um país ser da UE dado que ele está avançado em relação à métrica é de 95.7%.

```
[24]: # P(UE / Atrasado) = P(UE) * P(Atrasado / UE) / P(Atrasado)
p_ue_atrasado = (p_atrasado_ue * p_ue) / p_atrasado
print(f'De acordo com o Bayesiano calculado, a probabilidade de um país ser da
↳ UE dado que ele está atrasado em relação à métrica é de {p_ue_atrasado*100:.
↳ 1f}%.'
```

De acordo com o Bayesiano calculado, a probabilidade de um país ser da UE dado que ele está atrasado em relação à métrica é de 31.2%.

```
[25]: # Classificador Bayesiano
def classificador_bayesiano(pais):
    if df_concat[df_concat['Country Code'] == pais]['Status'].values[0] ==
↳ 'Avançado':
```

```

    if p_ms_avancado > p_ue_avancado:
        return 'Mercosul'
    else:
        return 'UE'
else:
    if p_ms_atrasado > p_ue_atrasado:
        return 'Mercosul'
    else:
        return 'UE'

```

6 Classificador Bayesiano

Para realizar a classificação simples, o grupo elaborou uma função classificadora com base nas probabilidades calculadas anteriormente.

```

[26]: df_concat['Pred'] = df_concat['Country Code'].apply(classificador_bayesiano)
df_concat.head()

```

```

[26]:
Country Name Country Code Indicator Name \
14 Austria AUT CO2 emissions (metric tons per capita)
17 Belgium BEL CO2 emissions (metric tons per capita)
21 Bulgaria BGR CO2 emissions (metric tons per capita)
53 Cyprus CYP CO2 emissions (metric tons per capita)
54 Czech Republic CZE CO2 emissions (metric tons per capita)

Indicator Code 2000 2001 2002 2003 2004 \
14 EN.ATM.CO2E.PC 7.929786 8.444109 8.583317 9.168344 9.275614
17 EN.ATM.CO2E.PC 11.439581 11.504321 10.727988 11.131315 10.941224
21 EN.ATM.CO2E.PC 5.314454 5.767160 5.563239 6.164371 6.125549
53 EN.ATM.CO2E.PC 7.558667 7.279172 7.339033 7.880736 7.699845
54 EN.ATM.CO2E.PC 12.010653 12.011818 11.624103 12.043361 12.105402

2005 ... 2014 2015 2016 2017 2018 \
14 9.266114 ... 7.260404 7.317159 7.288820 7.486162 7.133064
17 10.555782 ... 8.041711 8.437852 8.314932 8.151096 8.196339
21 6.265854 ... 5.820925 6.207865 5.834882 6.201290 5.822034
53 7.687390 ... 5.909935 5.900152 6.169943 6.238954 6.054175
54 11.750804 ... 9.264303 9.400668 9.627750 9.611647 9.492068

2019 Delta Status Bloco Pred
14 7.293984 -0.635801 Avançado UE UE
17 8.095584 -3.343997 Avançado UE UE
21 5.610857 0.296404 Atrasado UE Mercosul
53 5.998795 -1.559872 Avançado UE UE
54 9.022786 -2.987867 Avançado UE UE

```

[5 rows x 28 columns]

```
[27]: # Avaliando o classificador
df_concat['Pred'].value_counts()
```

```
[27]: UE          23
Mercosul      16
Name: Pred, dtype: int64
```

```
[28]: # Avaliando a contagem de um bloco previsto em relação ao bloco real dos países
df_concat.groupby(['Bloco', 'Pred']).size()
```

```
[28]: Bloco      Pred
Mercosul  Mercosul    11
         UE          1
UE        Mercosul    5
         UE          22
dtype: int64
```

```
[29]: # Acurácia do classificador
from sklearn.metrics import accuracy_score
acc = accuracy_score(df_concat['Bloco'], df_concat['Pred'])
print(f'A acurácia do classificador é de {acc*100:.2f}%.')

# Acurácia por bloco
df_concat.groupby(['Bloco', 'Pred']).size() / df_concat.groupby(['Bloco']).
↳size()
```

A acurácia do classificador é de 84.62%.

```
[29]: Bloco      Pred
Mercosul  Mercosul    0.916667
         UE          0.083333
UE        Mercosul    0.185185
         UE          0.814815
dtype: float64
```

7 Conclusão

É possível observar, pela matriz elaborada, que apenas um país da Mercosul foi erroneamente calculado como sendo da União Europeia, enquanto 5 países da União Europeia foram erroneamente calculados como sendo do Mercosul. A acurácia calculada para o classificador foi de quase 85%, o que é um resultado satisfatório para um classificador tão simples. A acurácia é medida contando os valores previstos como verdadeiros, divididos pelo total de valores previstos; portanto, em 85% dos casos, o classificador previu corretamente o bloco econômico ao qual o país pertence. Como pode ser visto em seguida, a acurácia por bloco é de aproximadamente 92% para o Mercosul, e 81% para a União Europeia. Isso provavelmente se dá por conta de haver mais países na União Europeia

do que no Mercosul, e que tais países são mais diversos em relação à métrica. O Mercosul possui menos países, e são mais concentrados em “Atrasados” no Status, como pode ser visto em nossa Análise Exploratória.

Levando em consideração o que foi obtido, o grupo pode concluir que, para essa situação, a métrica escolhida é um bom preditor para o bloco econômico.

7.1 Alterações na APS2

- Retiramos a afirmação de uma correlação negativa significar que as variáveis são inversamente proporcionais.
- Separamos os gráficos, para uma melhor compreensão do eixo Y, um está em %, enquanto o outro está em toneladas
- Mudamos a ordem de apresentação dos resultados, estando as conclusões por último, após os gráficos e os valores de covariância e correlação.

8 PARTE 4

8.1 Introdução

Para a APS4, foi pedido aos alunos elaborar uma reamostragem Bootstrap dos dados, tentando comprovar, com uma confiança de 90%, que realmente há diferença na probabilidade dos blocos pertencerem ao Status Avançado ou Atrasado em relação à métrica anteriormente estabelecida.

```
[30]: df_parte_4 = df_concat.copy()
      print(df_parte_4.shape)
      df_parte_4.head()
```

(39, 28)

```
[30]:
```

	Country Name	Country Code	Indicator Name	\
14	Austria	AUT	CO2 emissions (metric tons per capita)	
17	Belgium	BEL	CO2 emissions (metric tons per capita)	
21	Bulgaria	BGR	CO2 emissions (metric tons per capita)	
53	Cyprus	CYP	CO2 emissions (metric tons per capita)	
54	Czech Republic	CZE	CO2 emissions (metric tons per capita)	

	Indicator Code	2000	2001	2002	2003	2004	\
14	EN.ATM.CO2E.PC	7.929786	8.444109	8.583317	9.168344	9.275614	
17	EN.ATM.CO2E.PC	11.439581	11.504321	10.727988	11.131315	10.941224	
21	EN.ATM.CO2E.PC	5.314454	5.767160	5.563239	6.164371	6.125549	
53	EN.ATM.CO2E.PC	7.558667	7.279172	7.339033	7.880736	7.699845	
54	EN.ATM.CO2E.PC	12.010653	12.011818	11.624103	12.043361	12.105402	

	2005	...	2014	2015	2016	2017	2018	\
14	9.266114	...	7.260404	7.317159	7.288820	7.486162	7.133064	
17	10.555782	...	8.041711	8.437852	8.314932	8.151096	8.196339	

21	6.265854	...	5.820925	6.207865	5.834882	6.201290	5.822034
53	7.687390	...	5.909935	5.900152	6.169943	6.238954	6.054175
54	11.750804	...	9.264303	9.400668	9.627750	9.611647	9.492068

	2019	Delta	Status	Bloco	Pred
14	7.293984	-0.635801	Avançado	UE	UE
17	8.095584	-3.343997	Avançado	UE	UE
21	5.610857	0.296404	Atrasado	UE	Mercosul
53	5.998795	-1.559872	Avançado	UE	UE
54	9.022786	-2.987867	Avançado	UE	UE

[5 rows x 28 columns]

8.2 Calculando Probabilidades

Primeiramente, vamos calcular a probabilidade de um país pertencer ao Status Avançado ou Atrasado, dado o bloco econômico o qual participa, ou seja, $P(\text{Avançado} \mid \text{Bloco})$ e $P(\text{Atrasado} \mid \text{Bloco})$.

```
[31]: dfmercosul = df_parte_4[df_parte_4['Bloco'] == 'Mercosul']

p_atrasado_mercosul = dfmercosul[dfmercosul['Status'] == 'Atrasado'].shape[0] / \
    ↪dfmercosul.shape[0]
p_avancado_mercosul = dfmercosul[dfmercosul['Status'] == 'Avançado'].shape[0] / \
    ↪dfmercosul.shape[0]
print(f'Probabilidade de um país da Mercosul ser atrasado em relação à métrica: \
    ↪{p_atrasado_mercosul*100:.2f}%')
print(f'Probabilidade de um país da Mercosul ser avançado em relação à métrica: \
    ↪{p_avancado_mercosul*100:.2f}%')
```

Probabilidade de um país da Mercosul ser atrasado em relação à métrica: 91.67%
 Probabilidade de um país da Mercosul ser avançado em relação à métrica: 8.33%

```
[32]: dfue = df_parte_4[df_parte_4['Bloco'] == 'UE']

p_atrasado_ue = dfue[dfue['Status'] == 'Atrasado'].shape[0] / dfue.shape[0]
p_avancado_ue = dfue[dfue['Status'] == 'Avançado'].shape[0] / dfue.shape[0]
print(f'Probabilidade de um país da UE ser atrasado em relação à métrica: \
    ↪{p_atrasado_ue*100:.2f}%')
print(f'Probabilidade de um país da UE ser avançado em relação à métrica: \
    ↪{p_avancado_ue*100:.2f}%')
```

Probabilidade de um país da UE ser atrasado em relação à métrica: 18.52%
 Probabilidade de um país da UE ser avançado em relação à métrica: 81.48%

8.3 Análise por Bootstrap

Em seguida, partiremos para a reamostragem Bootstrap. Primeiramente, elaboramos uma função que calcula as probabilidades para cada bloco econômico. Em seguida, a função `bootstrap_probs()` realizará a amostragem com reposição para um bloco econômico, um número de iterações e um intervalo de confiança dados.

```
[33]: import numpy as np

def calc_probs(df, bloco):
    df_bloco = df[df['Bloco'] == bloco]
    p_atrasado = df_bloco[df_bloco['Status'] == 'Atrasado'].shape[0] / df_bloco.
    ↪shape[0]
    p_avancado = df_bloco[df_bloco['Status'] == 'Avançado'].shape[0] / df_bloco.
    ↪shape[0]
    return p_atrasado, p_avancado

def bootstrap_probs(df, bloco, n=1000, ci=0.9):
    probs = []
    for i in range(n):
        df_sample = df.sample(frac=1, replace=True)
        p_atrasado_i, p_avancado_i = calc_probs(df_sample, bloco)
        probs.append([p_atrasado_i, p_avancado_i])
    probs = np.array(probs)
    probs_mean = np.mean(probs, axis=0)
    probs_std = np.std(probs, axis=0)
    lower = (1-ci)/2
    upper = 1 - lower
    prob_ci_lower = np.quantile(probs, lower, axis=0)
    prob_ci_upper = np.quantile(probs, upper, axis=0)
    return probs_mean, probs_std, prob_ci_lower, prob_ci_upper

[42]: mercosul_mean, mercosul_std, mercosul_lower, mercosul_upper = ↪
    ↪bootstrap_probs(df_parte_4, 'Mercosul')
print('Probabilidades para o Mercosul: \n')
print(f'Probabilidade Média: Atrasado {mercosul_mean[0]*100:.2f}%, Avançado ↪
    ↪{mercosul_mean[1]*100:.2f}%')
print(f'Desvio Padrão: {mercosul_std[0]*100:.2f}%')
print(f'Intervalo de Confiança de 90% para a Probabilidade: \n'
      f'    Atrasado: {mercosul_lower[0]*100:.2f}% - {mercosul_upper[0]*100:.
    ↪2f}% \n'
      f'    Avançado: {mercosul_lower[1]*100:.2f}% - {mercosul_upper[1]*100:.
    ↪2f}%')
```

Probabilidades para o Mercosul:

Probabilidade Média: Atrasado 91.31%, Avançado 8.69%
Desvio Padrão: 8.58%

Intervalo de Confiança de 90% para a Probabilidade:

Atrasado: 76.40% - 100.00%

Avançado: 0.00% - 23.60%

```
[43]: ue_mean, ue_std, ue_lower, ue_upper = bootstrap_probs(df_parte_4, 'UE')
print('Probabilidades para a UE: \n')
print(f'Probabilidade Média: Atrasado {ue_mean[0]*100:.2f}%, Avançado{ue_mean[1]*100:.2f}%')
print(f'Desvio Padrão: {ue_std[0]*100:.2f}%')
print(f'Intervalo de Confiança de 90% para a Probabilidade: \n'
      f'    Atrasado: {ue_lower[0]*100:.2f}% - {ue_upper[0]*100:.2f}% \n'
      f'    Avançado: {ue_lower[1]*100:.2f}% - {ue_upper[1]*100:.2f}%')
```

Probabilidades para a UE:

Probabilidade Média: Atrasado 18.55%, Avançado 81.45%

Desvio Padrão: 7.65%

Intervalo de Confiança de 90% para a Probabilidade:

Atrasado: 7.14% - 32.00%

Avançado: 68.00% - 92.86%

8.4 Plotando os Resultados

Por fim, vamos colocar os resultados em um gráfico de barras para que consigamos enxergar melhor as probabilidades, seus intervalos de confiança e suas diferenças.

```
[44]: df_plot = pd.DataFrame({'Probabilidade': np.concatenate([mercosul_mean, ue_mean]),
                           'Bloco': ['Mercosul']*2 + ['UE']*2,
                           'Status': ['Atrasado', 'Avançado']*2,
                           'Confiança Inferior': np.concatenate([mercosul_lower, ue_lower]),
                           'Confiança Superior': np.concatenate([mercosul_upper, ue_upper])})

df_plot
```

	Probabilidade	Bloco	Status	Confiança Inferior	Confiança Superior
0	0.913138	Mercosul	Atrasado	0.763971	1.000000
1	0.086862	Mercosul	Avançado	0.000000	0.236029
2	0.185495	UE	Atrasado	0.071429	0.320000
3	0.814505	UE	Avançado	0.680000	0.928571

```
[45]: import matplotlib.pyplot as plt
import seaborn as sns
```



```

error_ms_av_0, error_ms_av_1 = df_plot['Confiança Superior'][1] -
    df_plot['Probabilidade'][1], df_plot['Probabilidade'][1] -
    df_plot['Confiança Inferior'][1]
error_ms_at_0, error_ms_at_1 = df_plot['Confiança Superior'][0] -
    df_plot['Probabilidade'][0], df_plot['Probabilidade'][0] -
    df_plot['Confiança Inferior'][0]
error_ue_av_0, error_ue_av_1 = df_plot['Confiança Superior'][3] -
    df_plot['Probabilidade'][3], df_plot['Probabilidade'][3] -
    df_plot['Confiança Inferior'][3]
error_ue_at_0, error_ue_at_1 = df_plot['Confiança Superior'][2] -
    df_plot['Probabilidade'][2], df_plot['Probabilidade'][2] -
    df_plot['Confiança Inferior'][2]

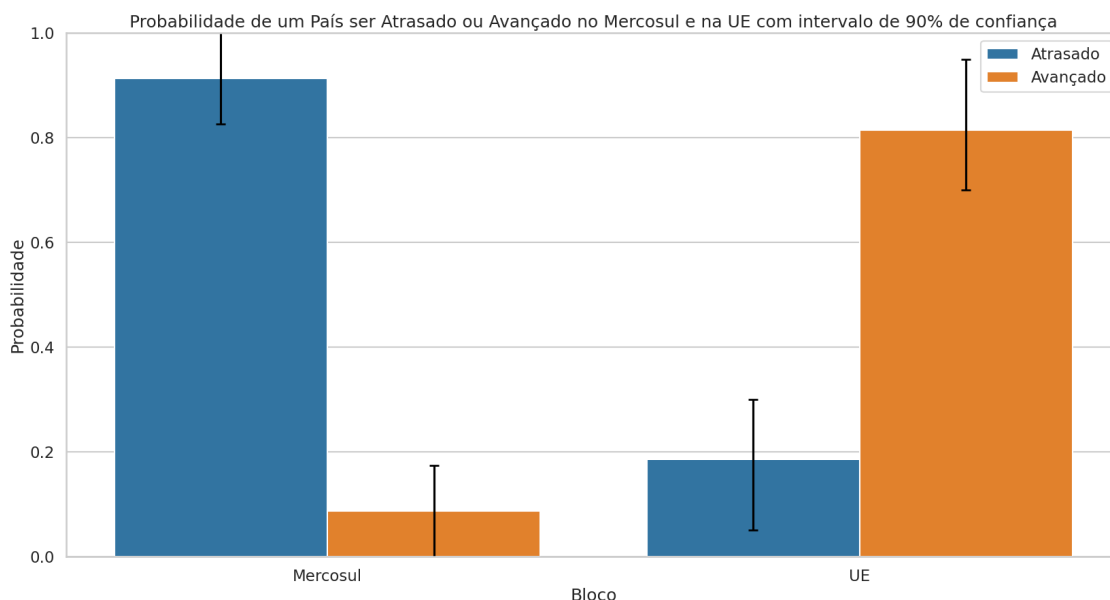
yerr = [[error_ms_at_0, error_ms_av_0, error_ue_at_0, error_ue_av_0],
        [error_ms_at_1, error_ms_av_1, error_ue_at_1, error_ue_av_1]]

```

```

[49]: plt.figure(figsize=(20,10))
sns.set_style('whitegrid')
sns.set_context('talk')
sns.barplot(data=df_plot, x='Bloco', y='Probabilidade', hue='Status')
plt.errorbar(x=[-0.2, 0.2, 0.8, 1.2], y=df_plot['Probabilidade'], yerr=yerr,
    fmt='none', ecolor='black', capsize=5, capthick=2)
plt.legend(loc='best')
plt.ylim(0, 1)
plt.ylabel('Probabilidade')
plt.xlabel('Bloco')
plt.title('Probabilidade de um País ser Atrasado ou Avançado no Mercosul e na
    UE com intervalo de 90% de confiança')
plt.show()

```



Como pode ser observado, não há a sobreposição de intervalos de confiança, o que indica que as probabilidades são realmente diferentes, com uma confiança de 90%.

8.5 Discussão e Conclusão

Com a observação dos gráficos e a análise dos resultados da amostragem Bootstrap, é possível concluir que, com uma confiança de 90%, os países da União Europeia são mais avançados em relação à métrica escolhida pelo grupo que países do Mercosul. Os intervalos de confiança não se sobrepõem, o que indica que as probabilidades realmente são diferentes para a confiança estipulada.

Mesmo com poucos dados para os países em análise, tal resultado provavelmente se dá pela alta discrepância entre os blocos econômicos escolhidos pelo grupo em relação à métrica: como pôde ser visto anteriormente, a probabilidade de um país da UE ser avançado em relação à métrica é de mais de 80%, enquanto a probabilidade de um país da Mercosul ser atrasado é de mais de 90%. Mesmo se reduzirmos o número de iterações na reamostragem bootstrap ou se aumentarmos o intervalo de confiança, o resultado se provará o mesmo, como pode ser visto a seguir, pois as diferenças entre os blocos são muito grandes.

```
[56]: mercosul_mean, mercosul_std, mercosul_lower, mercosul_upper =  
↳ bootstrap_probs(df_parte_4, 'Mercosul', 100, 0.95)  
print('Probabilidades para o Mercosul: \n')  
print(f'Probabilidade Média: Atrasado {mercosul_mean[0]*100:.2f}%, Avançado_  
↳ {mercosul_mean[1]*100:.2f}%')  
print(f'Desvio Padrão: {mercosul_std[0]*100:.2f}%')  
print(f'Intervalo de Confiança de 95% para a Probabilidade: \n'  
f'    Atrasado: {mercosul_lower[0]*100:.2f}% - {mercosul_upper[0]*100:.  
↳ 2f}% \n'  
f'    Avançado: {mercosul_lower[1]*100:.2f}% - {mercosul_upper[1]*100:.  
↳ 2f}%')
```

Probabilidades para o Mercosul:

Probabilidade Média: Atrasado 92.62%, Avançado 7.38%

Desvio Padrão: 7.84%

Intervalo de Confiança de 95% para a Probabilidade:

Atrasado: 73.81% - 100.00%

Avançado: 0.00% - 26.19%

```
[57]: ue_mean, ue_std, ue_lower, ue_upper = bootstrap_probs(df_parte_4, 'UE', 100, 0.  
↳ 95)  
print('Probabilidades para a UE: \n')  
print(f'Probabilidade Média: Atrasado {ue_mean[0]*100:.2f}%, Avançado_  
↳ {ue_mean[1]*100:.2f}%')  
print(f'Desvio Padrão: {ue_std[0]*100:.2f}%')  
print(f'Intervalo de Confiança de 95% para a Probabilidade: \n'  
f'    Atrasado: {ue_lower[0]*100:.2f}% - {ue_upper[0]*100:.2f}% \n'
```

f' **Avançado:** {ue_lower[1]*100:.2f}% - {ue_upper[1]*100:.2f}%')

Probabilidades para a UE:

Probabilidade Média: Atrasado 18.22%, Avançado 81.78%

Desvio Padrão: 7.40%

Intervalo de Confiança de 95% para a Probabilidade:

Atrasado: 6.90% - 34.09%

Avançado: 65.91% - 93.10%

Isso confirma o que foi dito anteriormente. Mesmo com as confianças maiores e número de reamostragens menores, os intervalos ainda não se sobrepõem, mantendo os resultados apresentados.

8.6 Mudanças na APS3

Não haviam mudanças a serem contempladas no feedback recebido.

8.7 Referências

- <https://www.mercosur.int/pt-br/quem-somos/paises-do-mercosul/>
- <https://www.mercosur.int/pt-br/quem-somos/em-poucas-palavras/>
- https://european-union.europa.eu/principles-countries-history/country-profiles_pt
- https://www.generaltradeways.com.br/es_ES/blog/o-que-e-o-mercosul-e-quais-os-seus-beneficios/
- https://pt.wikipedia.org/wiki/Uni%C3%A3o_Europeia
- https://pt.wikipedia.org/wiki/Crit%C3%A9rios_de_Copenhaga
- <https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c>