FUNDAMENTOS E TÉCNICAS EM CIÊNCIAS DE DADOS

PROF. JOSENALDE OLIVEIRA

josenalde@eaj.ufrn.br
https://github.com/josenalde/datascience

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UFRN

PANDAS #5 – AGREGAÇÃO E OPERAÇÕES EM GRUPOS

Classificar um conjunto de dados e aplicar funções a cada grupo, seja agregação ou uma transformação

Após carregar e mesclar conjuntos de dados, podem-se calcular estatísticas de grupo

Operação GROUPBY
Tabelas Pivô

Vamos olhar este dataset: https://www.kaggle.com/unsdsn/world-happiness

https://worldhappiness.report/

```
df = pd.read_csv('../datasets/wh2019.csv')
```

```
df.head()
```

Caso queira renomear colunas...

```
df.rename(columns={'Country or region': 'country'}, inplace=True)
```

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	1	Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393
1	2	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410
2	3	Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341
3	4	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118
4	5	Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298

Lista de países: `df['Country or region'].unique()`

PANDAS #5 – AGREGAÇÃO E OPERAÇÕES EM GRUPOS

Classificar um conjunto de dados e aplicar funções a cada grupo, seja agregação ou uma transformação

Após carregar e mesclar conjuntos de dados, podem-se calcular estatísticas de grupo

Operação GROUPBY
Tabelas Pivô

Vamos olhar este dataset: https://www.kaggle.com/unsdsn/world-happiness

https://worldhappiness.report/

```
df = pd.read_csv('../datasets/wh2019.csv')
```

```
df.head()
```

Caso queira renomear colunas...

```
df.rename(columns={'Country or region': 'country'}, inplace=True)
```

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	1	Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393
1	2	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410
2	3	Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341
3	4	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118
4	5	Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298

Lista de países: `df['Country or region'].unique()`

PANDAS #5 – AGREGAÇÃO E OPERAÇÕES EM GRUPOS

Como o de 2019 não tem a identificação de região, seria necessário um mapeamento para tal fim

Vamos olhar o relatório de 2016:

	Country	Region	Happiness Rank	Happiness Score	Lower Confidence Interval	Upper Confidence Interval	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Denmark	Western Europe	1	7.526	7.460	7.592	1.44178	1.16374	0.79504	0.57941	0.44453	0.36171	2.73939
1	Switzerland	Western Europe	2	7.509	7.428	7.590	1.52733	1.14524	0.86303	0.58557	0.41203	0.28083	2.69463
2	Iceland	Western Europe	3	7.501	7.333	7.669	1.42666	1.18326	0.86733	0.56624	0.14975	0.47678	2.83137
3	Norway	Western Europe	4	7.498	7.421	7.575	1.57744	1.12690	0.79579	0.59609	0.35776	0.37895	2.66465
4	Finland	Western Europe	5	7.413	7.351	7.475	1.40598	1.13464	0.81091	0.57104	0.41004	0.25492	2.82596

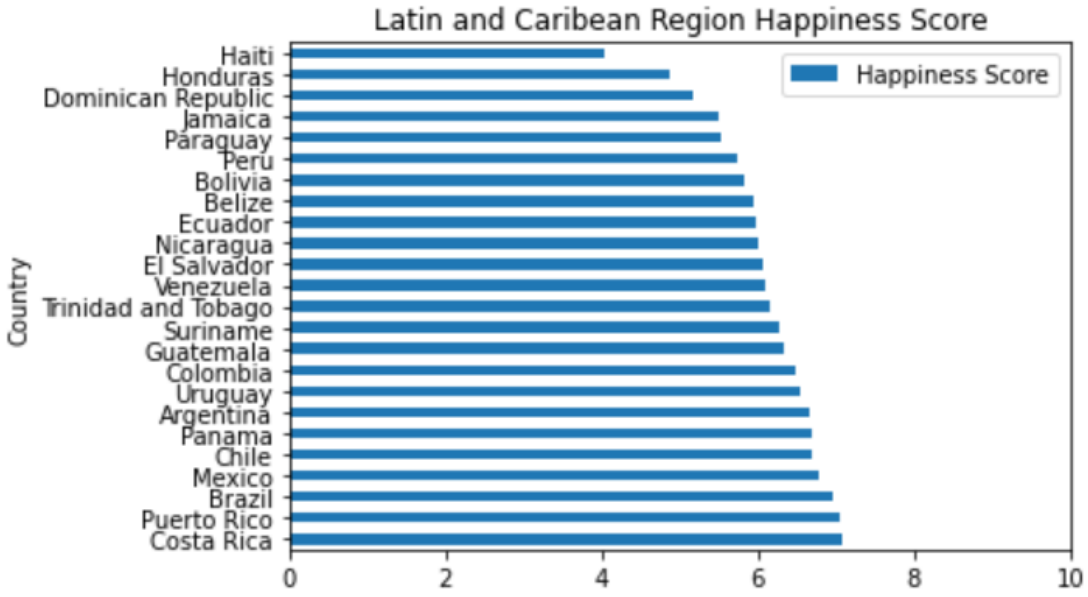
```
Objetivo:
1) plotar gráfico comparativo nos países da região Latin America and Caribean
2) plotar um gráfico do happiness score médio por região
```

PANDAS #5 – AGREGAÇÃO E OPERAÇÕES EM GRUPOS

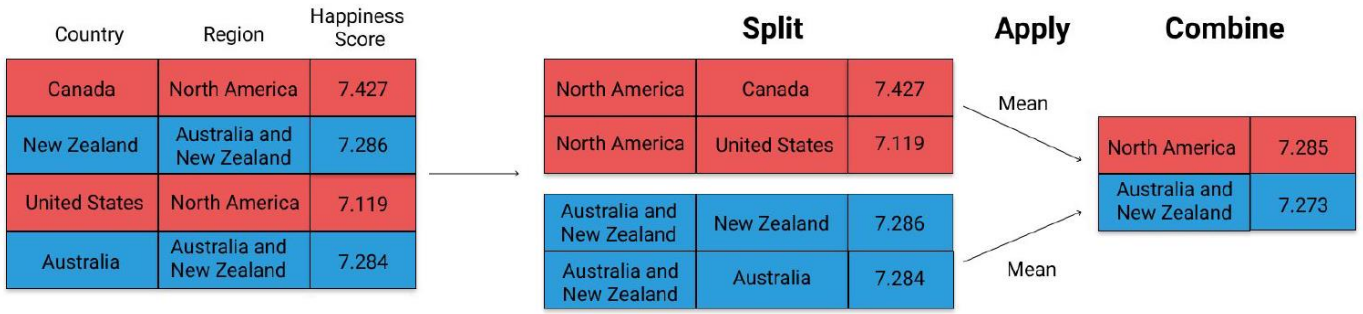
```
hs_latin = df2[df2['Region'].str.match(r'Latin.*')]
```

```
hs_latin.plot(x='Country',
              y='Happiness Score',
              kind='barh',
              title='Latin and Caribean Region Happiness Score',
              xlim=(0,10))
```

```
<AxesSubplot:title={'center':'Latin and Caribean Region Happiness Score'}, ylabel='Country'>
```



Entendendo a operação GroupBY



Fonte: slides Prof. Ivanovitch@IMD – lesson#7

PANDAS #5 – AGREGAÇÃO E OPERAÇÕES EM GRUPOS

1) Criar objeto GroupBy
2) Ter acesso ao número de grupos - ngroups
3) Obter itens num grupo E iterar sobre cada grupo
4) Aplicar funções de agregação

```
<pandas.core.groupby.generic.DataFrameGroupBy object at
0x00000169C674FEB0>
```

```
group_region = df2.groupby('Region')
```

```
group_region
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x00000169C674FEB0>
```

```
w_europe = group_region.get_group('Western Europe')
w_europe.head()
```

	Country	Region	Happiness Rank	Happiness Score	Lower Confidence Interval	Upper Confidence Interval	Economy (GDP per Capita)	Family	Health (Life Expectancy)	Freedom	Trust (Government Corruption)	Generosity	Dystopia Residual
0	Denmark	Western Europe	1	7.526	7.460	7.592	1.44178	1.16374	0.79504	0.57941			
1	Switzerland	Western Europe	2	7.509	7.428	7.590	1.52733	1.14524	0.86303	0.58557			
2	Iceland	Western Europe	3	7.501	7.333	7.669	1.42666	1.18326	0.86733	0.56624			

5) size() retorna o número de itens em cada grupo

Funções de agregação transformação

Methods	Description
mean()	Calculates the mean of groups.
sum()	Calculates the sum of group values.
size()	Calculates the size of the groups.
count()	Calculates the count of values in groups.
min()	Calculates the minimum of group values.
max()	Calculates the maximum of group values.

PANDAS #5 – AGREGAÇÃO E OPERAÇÕES EM GRUPOS

1) Aplicando sobre a coluna Happiness Score a média e o máximo

```
hs_grouped = group_region['Happiness Score']
hs_grouped.agg(['mean', 'max'])
```

Region	mean	max
Australia and New Zealand	7.323500	7.334
Central and Eastern Europe	5.370690	6.596
Eastern Asia	5.624167	6.379
Latin America and Caribbean	6.101750	7.087
Middle East and Northern Africa	5.386053	7.267
North America	7.254000	7.404
Southeastern Asia	5.338889	6.739
Southern Asia	4.563286	5.196
Sub-Saharan Africa	4.136421	5.648
Western Europe	6.685667	7.526

```
group_region_mean = df2.groupby(['Region'])['Happiness Score'].mean()
```

```
group_region_mean
```

```
Region
Australia and New Zealand          7.323500
Central and Eastern Europe         5.370690
Eastern Asia                       5.624167
Latin America and Caribbean        6.101750
Middle East and Northern Africa    5.386053
North America                      7.254000
Southeastern Asia
Southern Asia
Sub-Saharan Africa
Western Europe
Name: Happiness Score, dtype: fl
```

Função padrão

```
df2.pivot_table(values='Happiness Score', index='Region', aggfunc=np.mean)
```

Region	Happiness Score
Australia and New Zealand	7.323500
Central and Eastern Europe	5.370690
Eastern Asia	5.624167
Latin America and Caribbean	6.101750
Middle East and Northern Africa	5.386053
North America	7.254000
Southeastern Asia	5.338889
Southern Asia	4.563286
Sub-Saharan Africa	4.136421
Western Europe	6.685667

Tabela pivô – usada para sintetizar dados. Agrega Tabelas de acordo com uma ou mais chaves. Organiza os dados num retângulo com chaves de grupo nas linhas e nas colunas